June 1995

LIDS-TH-2304

Studies in Hybrid Systems:
Modeling, Analysis, and Control

Michael Stephen Branicky

Laboratory for Information and Decision Systems
MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

19980115 172

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE June 1995	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
Studies in Hybrid Systems: Modeling, Analysis and Control

5. FUNDING NUMBERS
F49620-86-C-0127

6. AUTHORS
Michael S. Branicky

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology

AFRL-SR-BL-TR-98-

$CO\backslash$

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
110 Duncan Avenue, Room B-115
Bolling Air Force Base, DC 20332-8080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
See attached.

DTIC QUALITY INSPECTED 2

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Studies in Hybrid Systems:
Modeling, Analysis, and Control

by

Michael Stephen Branicky

Submitted to the Department of Electrical Engineering and Computer Science
on 26 May 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Science in Electrical Engineering and Computer Science

Abstract

Complex systems typically possess a hierarchical structure, characterized by continuous-variable dynamics at the lowest level and logical decision-making at the highest. Virtually all control systems today perform computer-coded checks and issue logical as well as continuous-variable control commands. Such are "hybrid" systems.

Traditionally, the hybrid nature of these systems is suppressed by converting them into either purely discrete or continuous entities. Motivated by real-world problems, we introduce "hybrid systems" as interacting collections of dynamical systems, evolving on continuous-variable state spaces, and subject to continuous controls and discrete phenomena.

We identify the discrete phenomena that arise in hybrid systems and review previously proposed models. We propose a hybrid control model, coupling differential equations and automata, that encompasses them. Our unified model is natural for posing and solving hybrid analysis and control problems.

We discuss topological issues that arise in hybrid systems analysis. Then we compare the computational capabilities of analog, digital, and hybrid machines by proposing intuitive notions of analog machines simulating digital ones. We show that simple continuous systems possess the power of universal computation. Hybrid systems have further simulation capabilities. For instance, we settle the famous asynchronous arbiter problem in both continuous and hybrid settings. Further, we develop analysis tools for limit cycle existence, perturbation robustness, and stability. We analyze a hybrid control system, typically used in aircraft, that logically switches between two conventional controllers. Stability of such systems has previously only been tested using extensive simulation; we prove global asymptotic stability for a realistic set of cases. Our tools demonstrate robustness of this stability with respect to "continuation" of the logical function.

We systematize the notion of a hybrid system governed by a hybrid controller using an optimal control framework. We prove theoretical results that lead us to algorithms for synthesizing such hybrid controllers. In particular, we prove existence of optimal and near optimal controls and derive "generalized quasi-variational inequalities" that the associated value function satisfies. We outline algorithms for solving these inequalities, based on a generalized Bellman equation, impulse control algorithms, and linear programming. Several illustrative examples are solved. The synthesized optimal hybrid controllers verify engineering intuition.

Thesis Supervisor: Sanjoy K. Mitter
Title: Professor of Electrical Engineering

Thesis Reader: Munther A. Dahleh, Associate Professor of Electrical Engineering
Thesis Reader: Gunter Stein, Adjunct Professor of Electrical Engineering

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
77 Massachusetts Ave., Room 35-415
Cambridge, MA 02139-4307   U.S.A.

Phone: 617-253-2832
FAX: 617-258-8553
E-mail: branicky@lids.mit.edu

16 June 1995

SCEEE Management Office
1101 Massachusetts Avenue
St. Cloud, FL 34769
Attn: Kelly Brown

Dear Administrator:

On 9 June 1995 I graduated from the Massachusetts Institute of Technology with a Doctor of Science degree in the field of Electrical Engineering and Computer Science. My studies were aided by an Air Force Office of Scientific Research Graduate Fellowship (Contract F49620-86-C-0127/SBF861-0436, Subcontract S-789-000-056), which is greatly appreciated.

Enclosed are several items, in fulfillment of my AFOSR fellowship obligations:

- a detailed resume, including lists of publications and references
- MIT Doctoral thesis: *Studies in Hybrid Systems: Modeling, Analysis, and Control*
- copies of the following publications (detailed citations appear in the resume):
    1. *Universal Computation and Other Capabilities of Continuous and Hybrid Systems*
    2. *Continuity of ODE Solutions*
    3. *Reflex Control for Robot System Preservation, Reliability, and Autonomy*, with T. S. Wikman and W. S. Newman
    4. *A Unified Framework for Hybrid Control*, with V. S. Borkar and S. K. Mitter
    5. *Stability of Switched and Hybrid Systems*
    6. *Analog Computation with Continuous ODEs*
    7. *Analyzing Continuous Switching Systems: Theory and Examples*
    8. *Topology of Hybrid Systems*
    9. *Reflexive Collision Avoidance: A Generalized Approach*, with T. S. Wikman and W. S. Newman

If any questions or comments arise, you may contact me by any means listed on my resume.

Sincerely,

Dr. Michael S. Branicky

JUN 19 1995

MICHAEL S. BRANICKY

Home: PO Box 397205, Cambridge, MA 02139-7205   Tel. (617) 577–0182   Fax (617) 577–0182
Office: MIT, 77 Mass. Ave., 35-415, Cambridge, MA 02139-4307   Tel. (617) 253–2832   Fax (617) 258–8553
Email: branicky@lids.mit.edu

INTERESTS

▷ research and teaching at the interface of electrical engineering and computer science
▷ interests and experience lie in the areas of systems and control (particularly hybrid systems, intelligent control, nonlinear systems, optimization, and modeling) and artificial intelligence (including robotics, machine learning, neural networks, and computer vision)

EDUCATION

Massachusetts Institute of Technology, Cambridge, MA
▷ Sc.D., Electrical Engineering and Computer Science, June 1995
▷ Thesis: "Studies in Hybrid Systems: Modeling, Analysis, and Control"
▷ Advisor: Prof. Sanjoy K. Mitter
▷ Minor: Computer Science

Case Western Reserve University, Cleveland, OH
▷ M.S., Electrical Engineering and Applied Physics, January 1990
▷ Thesis: "Rapid Configuration Space Transforms for Real-Time Robotic Reflexes"
▷ Advisor: Prof. Wyatt S. Newman

Case Western Reserve University, Cleveland, OH
▷ B.S., Electrical Engineering and Applied Physics, May 1987

HONORS

▷ AFOSR Graduate Fellowship, NSF Graduate Fellowship
▷ *Sigma Xi*, Mortar Board, *Tau Beta Pi, Eta Kappa Nu*
▷ National Merit Scholar, Ohio Academic Scholar
▷ full-tuition undergraduate scholarship
▷ Emily M. Hills Poetry Prize (CWRU; First Place, university-wide)

GRADUATE COURSEWORK

MIT:
Dynamical Systems
Multivariable Control Systems I and II
Dynamics of Nonlinear Systems
Advanced Control System Design
Nonlinear Control (Harvard cross-registration)
Fundamentals of System Theory
Special Topics: Optimal Control
Special Topics: Intelligent Control
Stochastic Processes, Detection, and Estimation
Recursive Estimation
Motor Learning
Legged Locomotion in Robots and Animals
Machine Learning
Artificial Intelligence
Theory of Computation
Theory of Algorithms
Applied Probability
Analysis
Topology
Differential Geometry

CWRU:
Data Acquisition and Control
Robotics I and II
Intelligent Systems I and II
Computer Vision for Industrial Applications
Multiprocessing and Special Architectures

EXPERIENCE	**Research Fellow / Research Assistant**	*9/90 — present*

MIT Laboratory for Information and Decision Systems, Cambridge, MA
▷ developed tools for the analysis of hybrid systems
▷ identified phenomena, formulated a framework, derived theoretical results, and produced algorithms for control of hybrid systems

Grader *Fall 1993, Fall 1992*
MIT Electrical Engineering and Computer Science Dept., Cambridge, MA
▷ graded problem sets for Dynamic Systems and Introduction to Communication, Control, and Signal Processing

Research Associate *7/93 — 8/93*
NASA Ames Research Center, Mountain View, CA
▷ worked with Dr. George Meyer and Prof. Michael Heymann on modeling and control of flight vehicle management systems
▷ programmed Statemate simulations of such systems

Summer Research Fellow *6/90 — 8/90*
USAF Armstrong Aerospace Medical Research Lab, Wright-Patterson AFB, OH
▷ modeled and controlled a bilateral force-reflecting telerobot testbed system
▷ derived kinematics and control for an exoskeleton/robot pair, culminating in peg-in-the-hole experiments

Research Fellow *9/89 — 5/90*
MIT Artificial Intelligence Lab, Cambridge, MA
▷ experimented on and extended robotic task-level learning with Prof. Chris G. Atkeson

Research Assistant *8/88 — 8/89*
CWRU Ctr. for Automation and Intelligent Systems Research, Cleveland, OH
▷ developed artificial reflexes for automatic collision avoidance
▷ produced real-time hierarchical control, integrated with sensing

Systems Consultant *2/88 — 8/88*
Carter, Epner and Associates, Akron, OH for **Ohio Bell,** Cleveland, OH
▷ examined Digital Overlay Network (DON) potential for major markets
▷ devised a plan for DON project implementation

Teaching Assistant *8/87 — 5/88*
CWRU Electrical Engineering and Applied Physics Dept., Cleveland, OH
▷ organized and taught recitation sections in Electronic Circuits I and II
▷ prepared and graded homeworks, quizzes, exams; managed Motors Lab

Intern *Summer 1987*
Ohio Bell, Cleveland, OH
▷ gathered and evaluated data for strategic projects: Video Switching and DON
▷ installed and developed software applications

PROFESSIONAL
▷ IEEE member (since 1987)
▷ Engineer in Training (first step of Professional Engineer)
▷ Reviewer: *IEEE Trans. Robotics and Automation, IEEE Trans. Neural Networks, IEEE Trans. Automatic Control,* IEEE Conf. Decision and Control, American Control Conf.
▷ Co-Chair, Nonlinear Control II Session, 1994 American Control Conference

EXTRA-CURRICULARS
▷ *Zeta Psi* International Fraternity (secretary, tax secretary, treasurer, Supreme Council member, scholarship chair, community service chair, sergeant-at-arms)
▷ University Program Board, *Case Reserve Review*
▷ intramural bowling, football, racquetball, softball, volleyball, ultimate frisbee

JOURNAL PUBLICATIONS

▷ "Universal Computation and Other Capabilities of Continuous and Hybrid Systems," *Theoretical Computer Science*, Special Issue on Hybrid Systems, 138:1, pp. 67–100, 1995.

▷ "Continuity of ODE Solutions," *Applied Mathematics Letters*, 7:5, pp. 57–60, 1994.

▷ "Reflex Control for Robot System Preservation, Reliability, and Autonomy," with Thomas S. Wikman and Wyatt S. Newman, *Computers and Electrical Engineering*, Special Issue on Fault Tolerance and Robotics, 20:5, pp. 391–407, 1994.

▷ "Real-Time Configuration Space Transforms for Obstacle Avoidance," with Wyatt S. Newman, *Int. Journal of Robotics Research*, 10:6, pp. 650–667, 1991.

CONFERENCE PUBLICATIONS

▷ "A Unified Framework for Hybrid Control," with Vivek S. Borkar and Sanjoy K. Mitter, *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, pp. 4228–4234, Dec. 1994.

▷ "Stability of Switched and Hybrid Systems," *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, pp. 3498–3503, Dec. 1994.

▷ "Analog Computation with Continuous ODEs," *Proc. IEEE Workshop Physics and Computation*, Dallas, pp. 265–274, Nov. 1994.

▷ "Analyzing Continuous Switching Systems: Theory and Examples," *Proc. 1994 American Control Conf.*, Baltimore, pp. 3110–3114, June 1994.

▷ "Topology of Hybrid Systems," *Proc. 32nd IEEE Conf. Decision and Control*, pp. 2309–2314, Dec. 1993.

▷ "Reflexive Collision Avoidance: A Generalized Approach," with Thomas S. Wikman and Wyatt S. Newman, *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, pp. 3-31–3-36, May 1993.

▷ "A Hierarchical Algorithm for Neural Training and Control," with Theodore V. Theodosopoulos and Mitch M. Livstone, *Proc. 30th Annual Allerton Conf. Communication, Control, and Computing,* Urbana, IL, pp. 913–921, Sept. 1992.

▷ "Task-Level Learning: Experiments and Extensions," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 266–271, April 1991.

▷ "Rapid Computation of Configuration-Space Obstacles," with Wyatt S. Newman, *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, pp. 304–310, May 1990.

▷ "Experiments in Reflex Control for Industrial Manipulators," with Wyatt S. Newman, *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, pp. 266–271, May 1990.

OTHER PUBLICATIONS

▷ "Optimized Forward Kinematics for the MBA Exoskeleton and Partitioned Kinematics for the Merlin Robot," with S.J. Remis, Technical Report AAMRL-SR-90-517, Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, Nov. 1990.

▷ "Control and Modeling for Improved Human Sensory Feedback," *1990 LGFP/DoD Fellowship Summer Research Program Final Report*, Universal Energy Systems, Inc., 1990.

▷ 2 published abstracts; 7 intramural technical reports

REFERENCES   ▷ Prof. Sanjoy K. Mitter
Dept. of Electrical Engineering and Computer Science
77 Massachusetts Avenue, 35-304
Massachusetts Institute of Technology
Cambridge, MA 02139-4307

Tel. (617) 253–2160, Fax (617) 258–8553
Email: mitter@lids.mit.edu

▷ Prof. Wyatt S. Newman
Dept. of Electrical Engineering and Applied Physics
516 Glennan Building
Case Western Reserve University
Cleveland, OH 44106-7221

Tel. (216) 368–6432, Fax (216) 368–2668
Email: wsn@r2d2.eeap.cwru.edu

▷ Prof. Munther A. Dahleh
Dept. of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
77 Massachusetts Avenue, 35-402
Cambridge, MA 02139-4307

Tel. (617) 253–3892, Fax (617) 258–8553
Email: dahleh@lids.mit.edu

▷ Prof. Anil Nerode
Mathematical Sciences Institute
409 College Avenue
Cornell University
Ithaca, NY 14850

Tel. (607) 255–3577
Email: anil@math.cornell.edu

▷ Prof. Shankar Sastry
Dept. of Electrical Engineering and Computer Science
261 M Cory Hall
University of California
Berkeley, CA 94720-1770

Tel. (510) 642–1857, Fax (510) 642–1341
Email: sastry@robotics.eecs.berkeley.edu

▷ Prof. Michael Heymann
Dept. of Computer Science
Technion
Haifa 32000 Israel

Tel. 972-4-294163, Fax 972-4-294353
Email: heymann@cs.technion.ac.il

June 1995

LIDS-TH-2304

Studies in Hybrid Systems: Modeling, Analysis, and Control

Michael Stephen Branicky

This report is based on the unaltered thesis of Michael Stephen Branicky submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Science in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology in June 1995.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Studies in Hybrid Systems:
Modeling, Analysis, and Control

by

Michael Stephen Branicky

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Author...
Department of Electrical Engineering and Computer Science
26 May 1995

Certified by ....
Sanjoy K. Mitter
Professor of Electrical Engineering
Thesis Supervisor

Accepted by..
F. R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Studies in Hybrid Systems:
Modeling, Analysis, and Control
by
Michael Stephen Branicky

Submitted to the Department of Electrical Engineering and Computer Science
on 26 May 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Science in Electrical Engineering and Computer Science

Abstract

Complex systems typically possess a hierarchical structure, characterized by continuous-variable dynamics at the lowest level and logical decision-making at the highest. Virtually all control systems today perform computer-coded checks and issue logical as well as continuous-variable control commands. Such are "hybrid" systems.

Traditionally, the hybrid nature of these systems is suppressed by converting them into either purely discrete or continuous entities. Motivated by real-world problems, we introduce "hybrid systems" as interacting collections of dynamical systems, evolving on continuous-variable state spaces, and subject to continuous controls and discrete phenomena.

We identify the discrete phenomena that arise in hybrid systems and review previously proposed models. We propose a hybrid control model, coupling differential equations and automata, that encompasses them. Our unified model is natural for posing and solving hybrid analysis and control problems.

We discuss topological issues that arise in hybrid systems analysis. Then we compare the computational capabilities of analog, digital, and hybrid machines by proposing intuitive notions of analog machines simulating digital ones. We show that simple continuous systems possess the power of universal computation. Hybrid systems have further simulation capabilities. For instance, we settle the famous asynchronous arbiter problem in both continuous and hybrid settings. Further, we develop analysis tools for limit cycle existence, perturbation robustness, and stability. We analyze a hybrid control system, typically used in aircraft, that logically switches between two conventional controllers. Stability of such systems has previously only been tested using extensive simulation; we prove global asymptotic stability for a realistic set of cases. Our tools demonstrate robustness of this stability with respect to "continuation" of the logical function.

We systematize the notion of a hybrid system governed by a hybrid controller using an optimal control framework. We prove theoretical results that lead us to algorithms for synthesizing such hybrid controllers. In particular, we prove existence of optimal and near optimal controls and derive "generalized quasi-variational inequalities" that the associated value function satisfies. We outline algorithms for solving these inequalities, based on a generalized Bellman equation, impulse control algorithms, and linear programming. Several illustrative examples are solved. The synthesized optimal hybrid controllers verify engineering intuition.

Thesis Supervisor: Sanjoy K. Mitter
Title: Professor of Electrical Engineering

Thesis Reader: Munther A. Dahleh, Associate Professor of Electrical Engineering
Thesis Reader: Gunter Stein, Adjunct Professor of Electrical Engineering

Acknowledgments

No man is an island. And no research is done in a vacuum. My life and research at MIT have been influenced by, and have benefited from, my involvement with a number of extraordinary advisors, teachers, colleagues, and friends.

Foremost, I would like to express my deep gratitude to my thesis supervisor, Prof. Sanjoy Mitter. He has been an advisor and supporter, but most importantly an example of a true scholar. I admire and owe him for his broad knowledge, connectionist approach, and deep insight—as well as for his trust and perseverance. I also owe him for managing a lab where students can explore new directions and for creating opportunities for me to learn and grow as a researcher.

I also thank the other members of my thesis committee, Profs. Munther Dahleh and Gunter Stein. Prof. Stein was always quick to grasp what I had done, point out how tough the problems really are, and keep me grounded with real-life examples (including the max system which started it all). Prof. Dahleh offered helpful comments and assistance throughout; he is an example of passion and excellence in both teaching and research.

Further, I am particularly indebted to three people who might as well have been on my thesis committee. Foremost, Prof. Anil Nerode has been like a second thesis supervisor to me. He has long been there to listen and to offer sound (and quick!) advice on both technical and professional matters. Over the years Dr. Charles Rockland has given me precise feedback and pointed me toward the best references. The optimal control theory in the thesis grew directly out of a class taught by Profs. Mitter and Vivek Borkar. That work is joint with both of them, but was catalyzed by Prof. Borkar's eagerness to listen, quick mind, and profound problem-solving ability.

While at MIT, I have had the good fortune to be taught by the best minds in systems and control theory, including Profs. Athans, Bertsekas, Borkar, Brockett, Dahleh, Drake, Mitter, Sastry, Slotine, Tsitsiklis, Van der Velde, Verghese, Willsky, and Wyatt. Other notable teachers include Profs. Atkeson, Guillemin, Leiserson, Munkres, Raibert, Rivest, and Singer. My summer at NASA Ames with Dr. George Meyer and Prof. Michael Heymann was particularly enjoyable and instructive; it also focused me on the ultimate goals of hybrid systems research. I thank Profs. Peter Caines, Eduardo Sontag, John Wyatt, and Sandro Zampieri for influential technical discussions on hierarchical systems, analog computation, asynchronous arbiters, and analog simulation, respectively. The multiple Lyapunov functions theory was inspired by conversations with Prof. Wyatt Newman. Thanks also to Charlie Horn and Prof. Peter Ramadge who pointed me to the hybrid systems work of LIDS' own Hans Witsenhausen, which was certainly ahead of its time. Thanks must also go to Ted Theodosopoulos, Mitch Livstone, Venkatesh Saligrama, and Prof. Ramadge, who proofread parts of the thesis.

Further, I am indebted to the lab staff who generally put up with me and helped me get things done throughout the years. Thanks Kathleen O'Sullivan, Sheila Hegarty, Julie Stewart, Margaret Flaherty, Nancy Young, and Fifa Monserrate.

□ □ □

I have been at this academic game for some time now. From my days at the Artificial Intelligence Lab I am particularly indebted to the guidance of Profs. Chris Atkeson and Marc Raibert. The influence of my days at Case also still lingers. I am thankful to have maintained friendships and professional interaction with Prof. Wyatt Newman, Kamal Souccar, Vinay Krishnaswamy, Werner Tseng, Mike Vertal, Mark Dohring, and Thomas Wickman. I am also honored to thank my most influential teachers from younger days at St. Edward, with whom I still have the pleasure of staying in touch, namely, Brs. Joseph Chvala and Bennett Nettleton.

MIT has been mostly a place of work, work, work, with a little sleep and food thrown in for survival purposes only. It has been important in such a place to work with colleagues who have also become friends, especially Chiquin Xie and Zexiang Li from the AI lab and Ted Theodosopoulos, Mitch Livstone, Venkatesh Saligrama, Francesca Villoresi, Stefano Casadei, Nicola Elia, Sandro Zampieri, Pete Young, and Bill Irving from LIDS. And there were also plain, old-fashioned good friends, who made MIT an enjoyable place and made me feel like I had a portion of a real life, especially Amir R. Amir and Carl Livadas. Also contributing to good times were Tony Ezzat, Chris Hadjicostis, Antonios Eleftheriou, and the rest of the Hellenic crowd. Further, it is a privilege to know the Theodosopoulos family (Vasilios and Euthemia, Phil, and newest member, Patty) and Dr. Riyadh Amir.

Other LIDS colleagues I would like to acknowledge are Alan Chao, Ian Chen, Mike Daniel, Joel Douglas, Marcos Escobar, Jorge Goncalves, Seema Jaggi, Cedric Logan, Mark Luettgen, Wesley McDermott, Steve Patek, Andy Potvin, Sekhar Tatikonda, and Ben Van Roy. There are also friends and colleagues from other institutions that I would like to acknowledge, particularly Mireille Broucke, Dawn Tilbury, Datta Godbole, Akash Deshpande, and Aleks Gollu from Berkeley and Charlie Horn from Princeton.

Finally, I would like to thank the friends and family who have always been there, despite the ravages of time and distance. Thanks Erik Bewley, John Melchiorre, Paul Kwiat, Andy Serafini, Kent Nee, and Michael and Peggy Epner. Thanks Aunt Sally and Uncle Don Bewley. Thanks Lisa, Mom, and Dad. Without you this victory would be hollow, if at all possible.

To everyone, I hope our paths continue to cross through the years and wish you happiness and good fortune in all that you do. I dedicate this thesis to my mother and father, Diana and Stephen.

Michael S. Branicky
Cambridge, MA
21 May 1995

Contents

List of Figures

Chapter 1

Introduction to Hybrid Systems

In this chapter, we motivate a study of hybrid systems and examine real-world examples where they arise. We explain our theoretical paradigm for such a study and formally introduce the objects of that study: autonomous and controlled hybrid dynamical systems. We outline the thesis and summarize its contributions.

§1.1 INTRODUCTION

MOTIVATION

Motivated by biology and a study of complex systems, intelligent behavior is typically associated with a hierarchical structure. Such a hierarchy exhibits an increase in reaction time and abstraction with increasing level. In both natural and engineered systems the lowest level is usually characterized by continuous-variable dynamics and the highest by a logical decision-making mechanism. The interaction of these different levels, with their different types of information, leads to a "hybrid" system.

Many complicated control systems today (e.g., those for flight control, manufacturing systems, and transportation) have vast amounts of computer code at their highest level. More pervasively, programmable logic controllers are widely used in industrial process control. We also see that today's products incorporate logical decision-making into even the simplest control loops (e.g., embedded systems). Thus, virtually all control systems today issue continuous-variable controls and perform logical checks that determine the mode—and hence the control algorithms—the continuous-variable system is operating under at any given moment. As such, these "hybrid control" systems offer a challenging set of problems.

So, "hybrid" systems are certainly pervasive today. But they have been with us at least since the days of the relay. Traditionally, though, the hybrid nature of systems and controllers has been suppressed by converting them into either purely discrete or purely continuous entities. The reason is that science and engineering's *formal* modeling, analysis, and control "toolboxes" deal largely—and largely successfully—with these "pure" systems.

Engineers have pushed headlong into the application areas above. And the successes in flight control alone attest to the fact that it is possible to build highly complex, highly reliable systems. Yet ever more complex systems continue to arise (e.g., flight vehicle management and intelligent vehicle/highway systems). And the trend toward embedded systems is sure to continue.

It is time to focus on developing formal modeling, analysis, and control methodologies for "hybrid systems."

WHAT ARE HYBRID SYSTEMS?

Generalizing from the examples above, *hybrid systems* involve both continuous-valued and discrete variables. Their evolution is given by equations of motion that generally depend on all variables. In turn these equations contain mixtures of logic, discrete-valued or **digital** dynamics, and continuous-variable or **analog** dynamics. The continuous dynamics of such systems may be continuous-time, discrete-time, or mixed (sampled-data), but is generally given by differential equations. The discrete-variable dynamics of hybrid systems is generally governed by a **digital automaton**, or input-output transition system with a countable number of states. The continuous and discrete dynamics interact at "event" or "trigger" times when the continuous state hits certain prescribed sets in the continuous state space. See Figure 1-1.
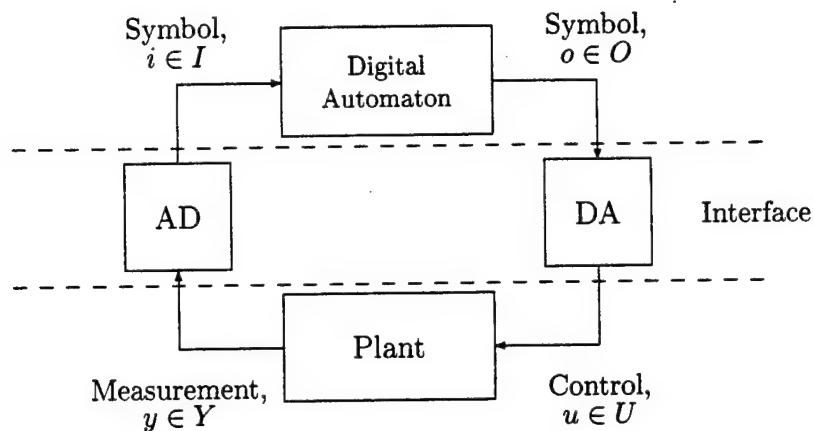
Figure 1-1: Hybrid System.

Hybrid control systems are control systems that involve both continuous and discrete dynamics and continuous and discrete controls. The continuous dynamics of such a system is usually modeled by a controlled vector field or difference equation. Its hybrid nature is expressed by a dependence on some discrete phenomena, corresponding to discrete states, dynamics, and controls. The result is a system as in Figure 1-2.
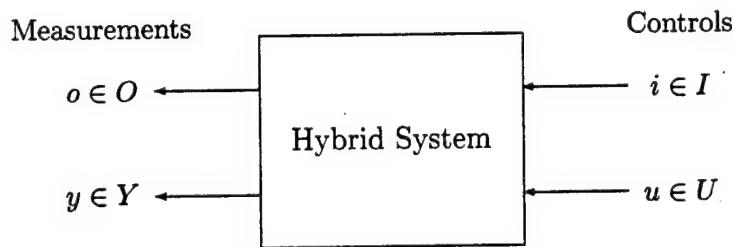
Figure 1-2: Hybrid Control System.

Below, we introduce **hybrid systems** as interacting collections of dynamical systems, each evolving on continuous state spaces, and subject to continuous and discrete controls, and some other discrete phenomena.

STUDIES IN HYBRID SYSTEMS

Research into such hybrid systems may be broken down into four broad categories:

- **Modeling**: formulating precise models that capture the rich behavior of hybrid systems.

  How do we "fill in the boxes" in Figures 1-1 and 1-2?

- **Analysis**: developing tools for the simulation, analysis, and verification of hybrid systems.

  How do we analyze systems as in Figure 1-1?

- **Control**: synthesizing hybrid controllers —which issue continuous controls and make discrete decisions—that achieve certain prescribed safety and performance goals for hybrid systems.

  How do we control a plant as in Figure 1-2 with a controller as in Figure 1-2?

- **Design**: conceiving new schemes and structures that lead to easier modeling, verification, and control of hybrid systems.

In this thesis, we concentrate on the first three categories, in Parts I, II, and III, respectively. Dependencies among the material is given in Figure 1-3.

Next, we give a quick overview of the thesis and its contributions. For more details consult §§1.5–1.7, in which we discuss modeling, analysis, and control contributions, in more depth.

Modeling. After examining the real-world examples in this chapter, we identify the discrete phenomena that generally arise in hybrid systems and give more examples. Then, we review in detail six previously posed hybrid systems models, primarily from the systems and control literature. We make some comparisons which enable us to later prove simulation and modeling results for such systems.

Throughout, we concentrate on the state-space and consider *general hybrid dynamical systems* as an indexed collection of dynamical systems along with some (deterministic, or *autonomous*) rules for "jumping" among them. These jumps take the form of switching dynamical systems and/or resetting their "continuous" states. This jumping generally occurs whenever the state satisfies certain conditions, given by its membership in a specified subset of the state space.

Controlled hybrid systems add the possibility of making discrete decisions at autonomous jump times as well the ability to discontinuously reset state variables at "intervention times" when the state satisfies certain conditions, given by its membership in another specified subset of the state space. In general, the allowed resettings depend on the state.

We introduce a hierarchy of such systems and provide a taxonomy of them based on their structure and the discrete phenomena they exhibit. We also give explicit instructions for computing the orbits and trajectories of general hybrid dynamical systems, including sufficient conditions for existence and uniqueness.

As a final result we introduce a precise model for hybrid control that is shown to encompass all identified phenomena and subsume all reviewed models. This "unified" model is suitable for the posing and solution of analysis and control problems in the sequel.

STUDIES IN HYBRID SYSTEMS
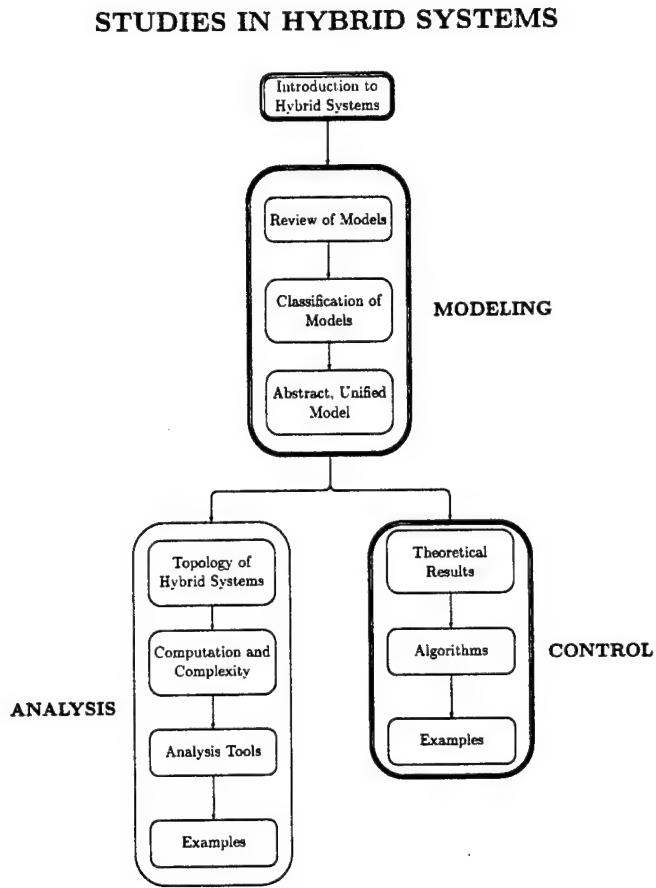
Figure 1-3: Dependency Structure of Thesis Topics.

Analysis.   We first discuss topological issues that arise in hybrid systems analysis. In particular, we examine topologies for achieving continuity of maps from a set of measurements of continuous dynamics to a finite set of input symbols (AD map of Figure 1-1) and back again (DA map). Finding some anomalies in completing this loop, we discuss a different view of hybrid systems that can broach them and is more in line with traditional control systems. The most widely used fuzzy control system is related to this different view and does not possess these anomalies. Indeed, we show that fuzzy control leads to continuous maps (from measurements to controls) and that all such continuous maps may be implemented via fuzzy control.

Then we compare the simulation and computational capabilities of analog, digital, and hybrid machines. We accomplish this by proposing several new, but intuitive, notions of simulation of a digital machine by an analog one. We show that even simple continuous systems, namely smooth ODEs in $\mathbb{R}^3$, possess the power of universal computation. And our simulation definitions *do not require infinite precision or precise timing.* Hybrid systems have further simulation capabilities. We contrast hybrid and continuous systems by solving the well-known arbiter problem in both a differential equations and a hybrid systems framework:

- You cannot build an arbiter with a set of Lipschitz ODEs with continuous output map.

- There is a simple (2-discrete-state, no continuous-state-jump) hybrid system—whose component ODEs are Lipschitz with continuous output map—that meets arbiter specifications. Each reviewed model can implement it.

Then we develop some tools for the analysis of hybrid systems. In particular, we develop two tools for stability of such systems:

- a new one which we call *multiple Lyapunov functions* for studying Lyapunov stability,

- iterated function systems (IFS) as a tool for Lagrange stability and positive invariance.

Other tools, such as an extension of Bendixson's Theorem for detecting limit cycles of certain hybrid systems and a robustness lemma for differential equations, are also developed. These tools have applications beyond the scope of hybrid systems.

Finally, our attention focuses on example systems and their analysis. The example systems, denoted *max systems*, arise from a realistic aircraft control problem which logically switches between two controllers (one for tracking and one for regulation about a fixed angle of attack) in order to achieve both reasonable performance and safety. While stability of such systems has previously only been examined using extensive simulation [134], we are able to prove global asymptotic stability for a realistic class of cases. Using our robustness lemma to compare ODE solutions, we extend the result to a class of "continuations" of max systems, which dynamically smooth the logical nonlinearity.

Control.   We systematize the notion of a hybrid system governed by a hybrid controller using an optimal control framework. We prove theoretical results that lead us to algorithms for synthesizing such hybrid controllers.

In particular, we define an optimal control problem in our unified hybrid control framework and derives some theoretical results. The problem, and all assumptions used in obtaining the remaining results, are expressly stated. Further, the necessity of these assumptions—or ones like them—is demonstrated. The main results are as follows:

- We prove the existence of optimal and near optimal controls.

- We derive "generalized quasi-variational inequalities" (GQVIs) that the associated value function is expected to satisfy.

Using the GQVIs as a starting point, we concentrate on algorithms for solving hybrid control problems. Our unified view led to the concept of examining a "generalized Bellman equation." We also draw explicit relations with impulse control of piecewise-deterministic processes. Four algorithmic approaches are outlined:

- an explicit boundary-value algorithm,

- generalized value iteration and policy iteration,

- modified impulse control algorithms,

- linear programming.

Finally, three illustrative examples are solved in our framework. We consider

- a hysteresis system that exhibits autonomous switching and has a continuous control;

- a satellite station-keeping problem involving controlled switching;

- a transmission problem with continuous accelerator input and discrete gear-shift position. In each case, the optimal controls produced verify engineering intuition.

Conclusion.   §12 concludes with a summary of contributions and a list of some open issues and future directions, including some words on **Design**.

AN INTRODUCTION ...

There has been much recent interest in studying hybrid systems [3, 5, 9, 32, 38, 56, 63, 64, 65, 66, 114, 120]. This chapter is both an introduction to that field and to this thesis. It is organized as follows.

We have already informally defined hybrid systems. In the next section,[1] §1.2, we provide real-world examples of hybrid systems. These serve the dual purpose of illustration and motivation. Next, we give a short history and examine approaches to the studies of hybrid systems. One is the paradigm adopted here: a hybrid system as *an interacting collection of dynamical systems.*

Thus, in §1.4 we are led to review the notion of a general dynamical system. This sets the stage for us to formally define the objects of our study: *hybrid dynamical systems.* Since we are interested in control, we introduce both autonomous and controlled versions.

We have already outlined our contributions above. In §§1.5–1.7, we consecutively discuss modeling, analysis, and control contributions in more depth.

Throughout the thesis we assume some familiarity with control theory [96, 133], differential equations [73], automata theory [20, 76], and topology [62, 75, 113]. Some review of notation and material is done in §2.1 and §A. A majority of the notation is collected in the Symbol Index, pp. 193–194. A general Index is also provided for convenience.

---

[1] The symbol § is used to cross-reference thesis sections.

§1.2 REAL-WORLD EXAMPLES OF HYBRID SYSTEMS

The prototypical hybrid systems are digital controllers, computers, and subsystems modeled as finite automata coupled with controllers and plants modeled by partial or ordinary differential equations or difference equations. Thus, such systems arise whenever one mixes logical decision-making with the generation of continuous control laws. More specifically, real-world examples of hybrid systems include

- systems with relays, switches, and hysteresis [135, 152],

- computer disk drives [65],

- transmissions, stepper motors, and other motion controllers [38],

- constrained robotic systems [9],

- intelligent vehicle/highway systems (IVHS) [64, 140],

- modern flexible manufacturing and flight control systems [80, 103].

Other important application areas for hybrid systems theory include embedded systems and analog/digital circuit co-design and verification.

We now briefly examine each of the above examples in more detail in turn.

Systems with Switches and Relays.   Physical systems with switches and relays can be naturally modeled as hybrid systems. Sometimes, the dynamics may be considered merely discontinuous, such as in a blown fuse. In many cases of interest, however, the switching mechanism has some hysteresis, yielding a discrete state on which the dynamics depends. This situation is depicted by the multi-valued function H shown in Figure 1-4.
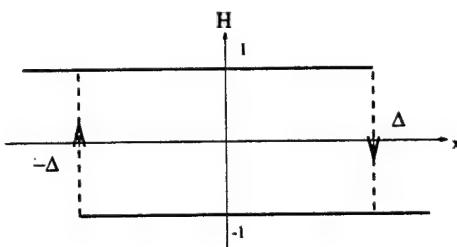
Figure 1-4: Hysteresis Function.

Suppose the function H models the hysteretic behavior of a thermostat. We may model a thermostatically controlled room as follows

$$\dot{x} = f(x, H(x - x_0)),\tag{1.1}$$

where x and x_0 denote room and desired temperature, respectively. The function f denotes dynamics of temperature, which depends on the current temperature and whether the furnace is switched On or Off. Note that this system is not just a differential equation whose right-hand side is piecewise continuous. There is "memory" in the system, which affects the value of the vector field. Indeed, such a system naturally has a finite automaton associated with the hysteresis function H, as pictured in Figure 1-5. Notice that, for example, the

discrete state changes from $+1$ to -1 when the continuous state enters the set $\{x \geq \Delta\}$. That is, the event of x attaining a value greater than or equal to Δ triggers the discrete or **phase** transition of the underlying automaton.
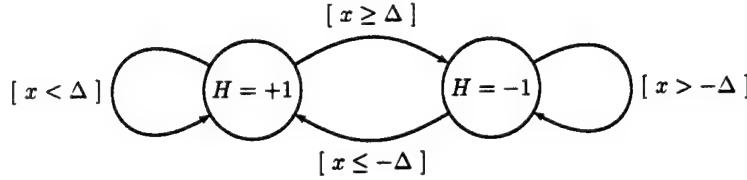
Figure 1-5: Finite Automaton Associated with Hysteresis Function.

Disk Drive. A computer disk drive may be modeled as a black box that receives external Read commands and outputs bytes. The action of the disk drive is governed by the differential (or difference equations) modeling the dynamic behavior of the disk, spindle, disk arm, and motors. The drive receives symbolic inputs of disk sectors and locations; it transmits symbolic outputs corresponding to the bytes read. It may also receive symbolic commands like Reinitialize and transmit symbolic outputs like ReadError. See Figure 1-6, which is constructed from the description in [65].
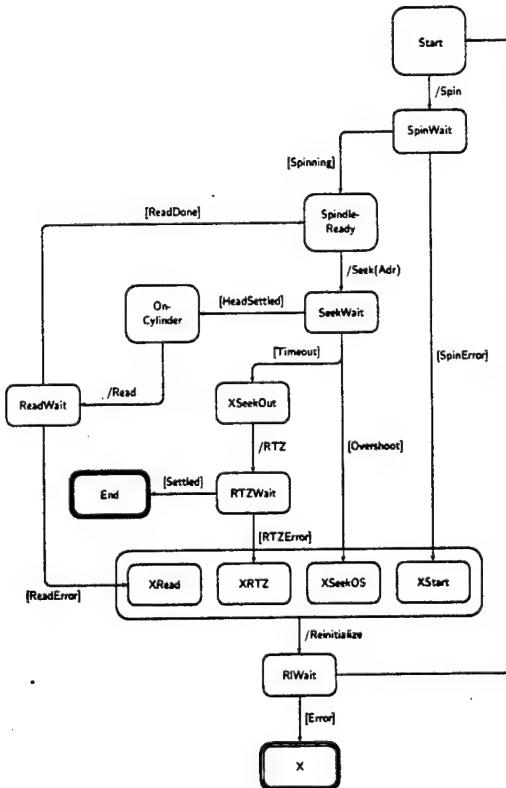
Figure 1-6: Finite State Machine Associated with Disk Drive Activities.

Transmission.   An automobile transmission system takes the continuous inputs of accelerator position and engine RPM and the discrete input of gear position and translates them into motion of the vehicle. Suppose one is designing a cruise control system that accelerates and decelerates under different profiles. The desired profile is chosen depending on sensor readings (e.g., continuous reading of elevation, discrete coding of road condition, etc.). In such a case, we are to design a control system with both continuous and discrete states and controls. See Example 3.4, p. 54.

Hopping Robot.   Interesting examples of hybrid systems are constrained robotic systems. In particular, consider the hopping robots of Marc Raibert of MIT [122]. The dynamics of these devices are governed by gravity, as well as the forces generated by passive and active (pneumatic) springs. The dynamics change abruptly at certain event times, and fall into distinct phases: Flight, Compression, Thrust, and Decompression. See Figure 1-7. In fact, Raibert has built controllers for these machines that embed a finite state machine that transitions according to these detected phases. For instance, the transition from Flight to Compression occurs when touchdown is detected; that from Decompression to Flight upon liftoff. Thus, finite automata and differential equations naturally interact in such devices and their controllers.

Flight   Compression

Thrust   Decompression
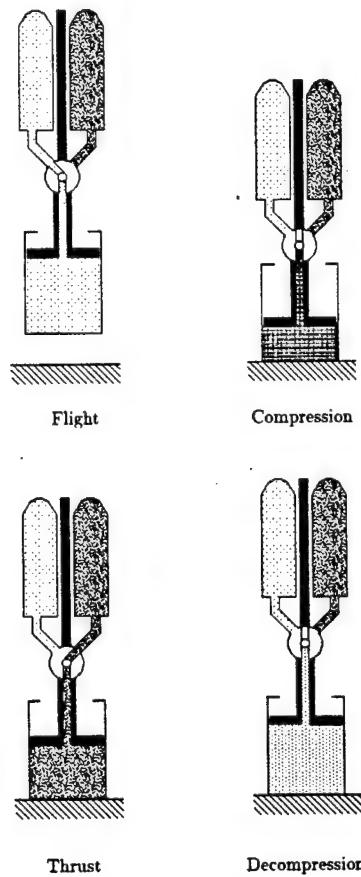
Figure 1-7: The dynamic phases of Raibert's hopping robot. Reproduced from [9].

IVHS. A more complicated example of a hybrid system arises in the control structures for so-called intelligent vehicle and highway systems (IVHS) [106, 140]. The basic goal of one such system is to increase highway throughput by means of a technique known as **platooning**. A platoon is a group of between, say, one and twenty vehicles traveling closely together in a highway lane at high speeds. To ensure safety—and proper formation and dissolution of structured platoons from the "free agents" of single vehicles—requires a bit of control effort! As in the theory of communication networks, researchers have broken this control task into layers [140]. See Figure 1-8.
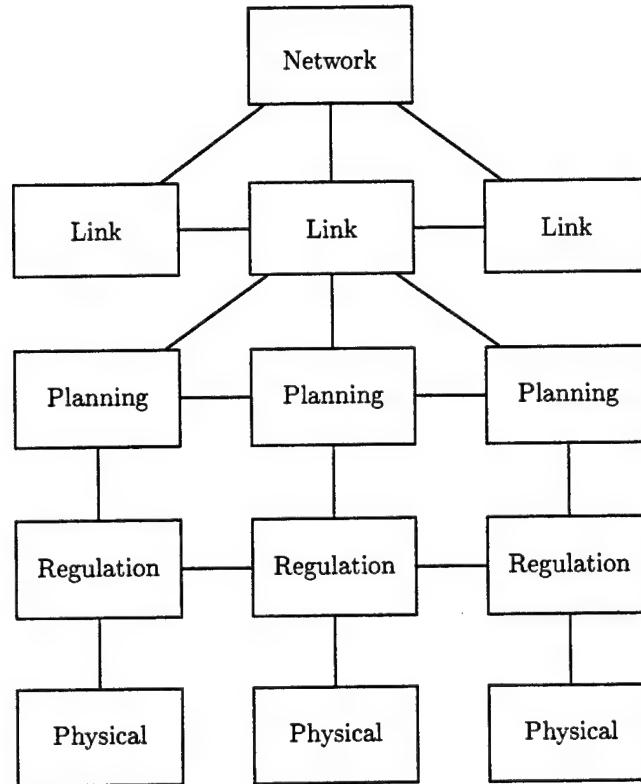
Figure 1-8: IVHS Control System Architecture.

Protocols for basic maneuvers such as Merge, Split, and ChangeLane have been proposed in terms of finite state machines. More conventional controllers govern the engines and brakes of individual vehicles. Clearly, the system is hybrid. Each vehicle has a state determined by:

- continuous variables, such as velocity, engine RPM, distance to car ahead,

- the finite state of its protocol-enacting automata.

The more conventional controllers can be analyzed for good performance using methods in control theory. The protocol designs can be verified using tools such as AT&T's COSPAN [72]. See Figure 1-9. The challenge with IVHS is to analyze the interconnected system as a whole: One has to verify (by proof or simulation) that the vehicles enact the protocol correctly and safely, under a range of dynamics conditions, for each of the possible product of finite states, for a wide scope of scenarios.
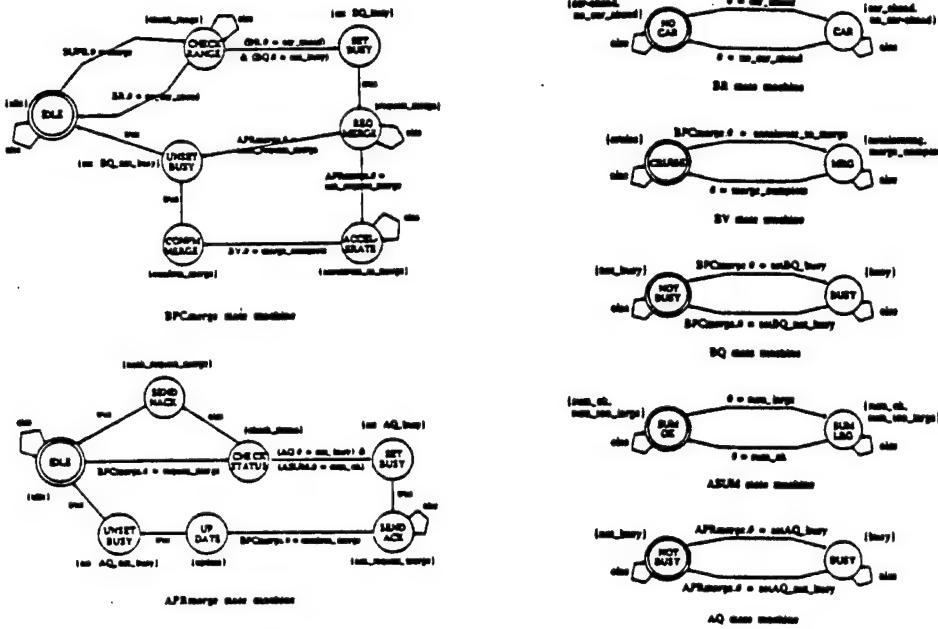
Figure 1-9: COSPAN specification of IVHS Merge protocol. Reproduced from [140].

Complex Systems. In the case of modern flexible manufacturing or flight control systems, there are typically many control subsystems with many control modes for each subsystem. The control subsystems that are active and the modes that are enforced at any given time is usually determined by a computer program. This computer program is a complex dynamical system in its own right and may be modeled as a finite automaton, pushdown automaton, Petri net, Turing machine, etc. The "transitions" of the computer program are not entirely independent of the physical system; many depend on the logical truth of statements concerning the continuous values of the physical variables. Since the logical state of the computer program determines which control mode is in use, the evolution of states of the physical system is likewise influenced by the values of these logical variables. In real-world systems, this interaction is complicated by the fact that there is not a strict dichotomy between discrete or logical components and continuous or physical components. The entire system is made up of subsystems that we choose to model as discrete, continuous, or hybrid. Figure 1-10 depicts a course-grained model of a modern flight vehicle management system (FVMS) proposed for the High-Speed Civil Transport [80, 103].

Interactions among the subsystems of a FVMS are typically so complicated that the only course of action in analyzing such systems is exhaustive simulation. This is currently done using programs such as Statemate [69, 70, 71, 81]. See Figure 1-11. The "statechart" pictured represents a simplified mode controller. It switches from the nominal mode of "pitch controlling climb, throttle controlling speed" to modes of "pitch controlling speed, climb clamped high/low" when the throttle exceeds upper and lower limits, respectively. There is also a "manual" mode, corresponding to the autopilot's being turned off.

In §9, we analyze a closely related aircraft control system, termed the *max system*, which switches between two different controllers in order to achieve a good tradeoff between performance and safety constraints. See Figure 9-2, p. 139.
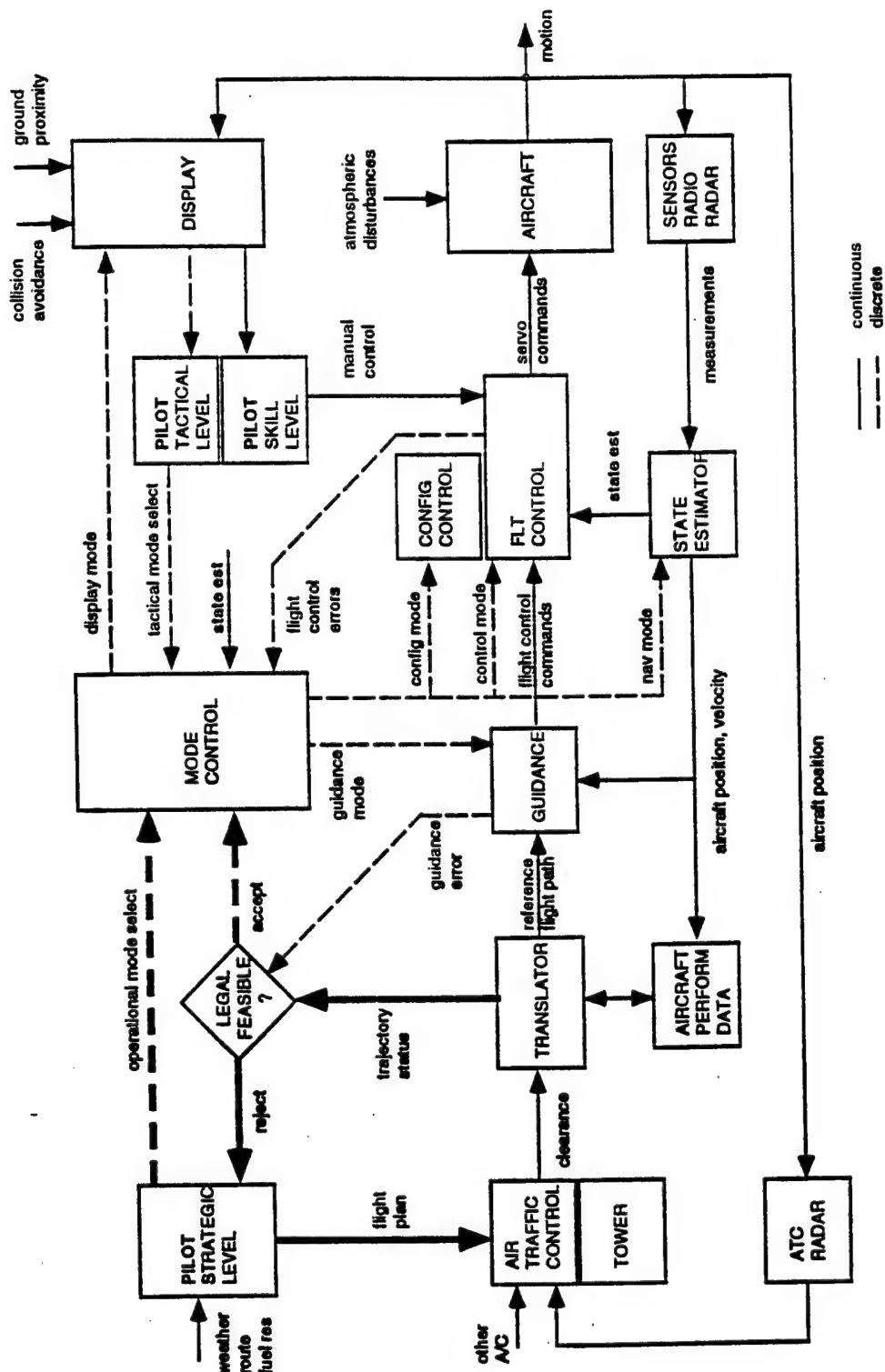
Figure 1-10: Flight Vehicle Management System architecture. Reproduced from [80].
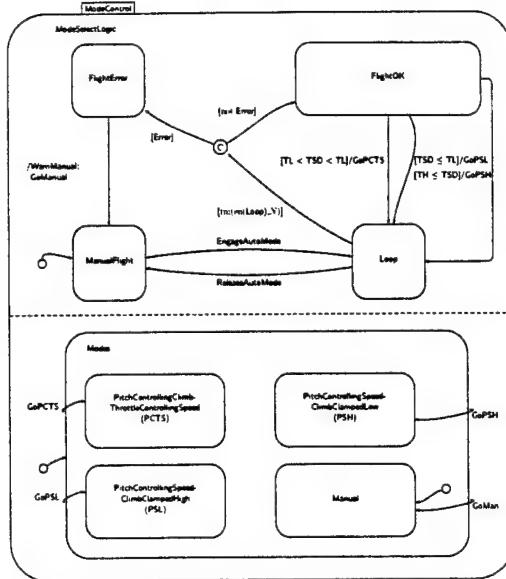
Figure 1-11: Statechart Description of a Simplified FVMS Mode Controller.

§1.3 STUDIES IN HYBRID SYSTEMS

§1.3.1 HISTORY IN BRIEF

Hybrid systems are certainly pervasive today. But they have been with us at least since the days of the relay. The earliest direct reference we know of is the visionary work of Hans Witsenhausen from MIT, who formulated a class of hybrid-state, continuous-time dynamic systems and examined an optimal control problem [152, §3.3, §10.5]. This was followed by others, like Pavlidis, who studied stability of systems with impulses via Lyapunov functions [118, §8.8]. Other early work on hybrid systems also came from MIT [83, 137, 151], where the interest was in finite state controllers. Ezzine and Haddad examined stability, controllability, and observability of a restricted class of switched linear systems [60, §2.2.2]. Motivated by an interest in systems with hysteresis, Tavernini produced a precise hybrid systems model and proved results on initial-value problems and their numerical approximations [135, §3.4].

In control theory, there has certainly been a lot of related work in the past, including variable structure systems, jump linear systems, systems with impulse effect, impulse control, and piecewise deterministic processes. These are quickly reviewed in §2.2. In computer science, there has been a successive build-up in the large formal verification literature [15, 54, 68, 84, 99, 146] toward verification of systems that include both continuous and discrete variables [3, 16, 66, 98, 120].

Recently, we have witnessed a resurgence in examining quantization effects [55, 89, 123, 128] and a heightened interest in analog computation [19, 27, 37, 46, 131, §7]. Finally, there has also been recent progress in analyzing switched [30, 93, 119, 149, §4.5, §8], hierarchical [40, 146], and discretely-controlled continuous-variable systems [41, 90, 112, 110, 111].

Hybrid systems have just started to be addressed more wholeheartedly by the control community [3, 5, 9, 32, 38, 56, 63, 64, 65, 66, 114, 120]. Computer scientists have also begun to attack this area [3, 66, 120].

§1.3.2 PARADIGMS

We see four basic paradigms for the study of hybrid systems: aggregation, continuation, automatization, and systemization. The first two approaches deal with the different sides—analog and digital—of hybrid systems. They attempt to suppress the hybrid nature of the system by converting it into a purely discrete or purely continuous one, respectively. The last two approaches are both more general and potentially more powerful. Under them, a hybrid system is seen directly as an interacting set of automata or dynamical systems; they complement the input-output and state-space paradigms, respectively, of both control theory and computer science. More specifically, the approaches are as follows.

1. **Aggregation** That is, suppress the continuous dynamics so that the hybrid system is a finite automaton or discrete-event dynamical system [74]. This is the approach most often taken in the literature, e.g., [5]. The drawback of this approach is three-fold.

   - *Nondeterminism*, i.e., one usually obtains a nondeterministic automaton. This was noted by Antsaklis, Stiver, and Lemmon [5]. Also cf. Hsu's cell-to-cell mapping [78].

   - *Nonexistence*, i.e., even if clever constructions are used, no finite automaton may exist that captures the combined behavior [63].

   - *Partition Problem*. It appears a conceptually deep problem to determine when there exist partitions of just a continuous system such that its dynamics is captured by a meaningful finite automaton. "Meaningful," since we note that every system is homomorphic to one with a single equilibrium point [129, §2.1.2]. The answer thus depends on the dynamical behavior one is interested in capturing and the questions one is asking. Readers interested in pursuing this topic should consult work on analog simulation of digital machines [27, 31, 36] as well as the more recent work of Prof. Roger Brockett [39].

   The aggregation program has been fully carried out so far only under strong assumptions on the hybrid system [1, 63].

2. **Continuation**, the complement of aggregation, that is, suppress the discrete dynamics so that the hybrid system becomes a differential equation. This original idea of Prof. Sanjoy Mitter and the author is to convert hybrid models into purely continuous ones—modeled by differential equations—using differential equations that simulate finite automata. In this familiar, unified realm one could answer questions of stability, controllability, and observability, converting them back to the original model by taking a "singular limit." For instance, one would like tools that allow one to conclude the following: if a "sufficiently close" continuation of a system is stable, then the original system is stable. Such a program is possible in light of the existence of simple continuations of finite automata [31, 36] and pushdown automata and Turing machines [31]. The drawback of this approach is three-fold.

   - *Arbitrariness*, i.e., how one accomplishes the continuation is largely arbitrary. For example, to interpolate or "simulate" the step-by-step behavior of a finite automaton Brockett used his double-bracket equations [38] and the author used stable linear equations [22, 31]. In certain cases this freedom is an advantage [§7]. However, care must be taken to insure that the dynamics used does not

introduce spurious behavior (like unwanted equilibria) or that it itself is not hard to analyze or predict.

- *Hiding Complexity*. One cannot generally get rid of the underlying discrete dynamics, i.e., the complexity is merely hidden in the "right-hand side" of the continuation differential equations [27].

- *Artificiality*. It can lead to a possibly unnatural analytical loop of going from discrete to continuous and back to discrete. Cf. Chen's recent results in stochastic approximation vis-à-vis Kushner's [42, 43].

The combination of these points has been borne out by some experience: it can be easier to examine the mixed discrete-continuous system. Cf. our analysis of a switched aircraft controller [28, §9] and Megretsky's recent analysis of a relay system [102].

3. **Automatization** or **automata approach**. Treat the constituent systems as a network of interacting automata [114, p. 325]. The focus is on the input-output or language behavior [20, 76]. The language view has been largely taken in the computer science literature in extending the dynamical behavior of finite automata incrementally toward full hybrid systems (see [1, 66] for background).

Automatization was pioneered in full generality by Nerode and Kohn [114]. The viewpoint is that systems, whether analog or digital, are automata. As long as there is compatibility between output and input alphabets, links between automata can be established. However, there is still the notion of "reconciling different time scales" [114, p. 325]. For instance, a finite automaton receives symbols in abstract time, whereas a differential equation receives inputs in "real time." This reconciliation can take place by either of the following:

- forcing synchronization at regular sampling instants [114, p. 333],

- synchronizing the *digital* automaton to advance at event times when its input symbols change [114, §3.2.5].

For hybrid systems of interest, the latter mechanism appears more useful. It has been used in many hybrid systems models, e.g., [5, 38, §§3.7–3.8]. It is reviewed in §3.6.

The automata approach has been taken most fruitfully by Deshpande [56].

4. **Systemization** or **systems approach**. Treat the constituent systems as interacting dynamical systems [129, §1.4, §2.1.2]. The focus is on the state-space [117]. The state-space view has been taken most profitably in the work of Witsenhausen [152] and Tavernini [135].

Systemization is developed in full generality in this thesis. The viewpoint is that systems, whether analog or digital, are dynamical systems. As long as there is compatibility at switching times when the behavior of a system changes in response to a logical decision or event occurrence, links between these dynamical systems can be established. Again, there is still the notion of reconciling dynamical systems with different time scales (i.e., transition semigroups). For instance, a finite automaton abstractly evolves on the positive integers (or on the free monoid generated by its input alphabet), whereas a differential equation evolves on the reals. This reconciliation can take place by either or both of the following:

- sequentially synchronizing the dynamical systems at event times when their states enter prescribed sets,

- forcing uniform semigroup structure via "timing maps."

Both approaches are introduced here, but the concentration is on the former.

Systemization is established in our formulation of hybrid dynamical systems below [§1.4,§5]. It is used in examining complexity and simulation capabilities of hybrid systems [31, §7], analyzing the stability of hybrid systems [28, 30, §§8–9], and in establishing the first comprehensive state-space paradigm for the control of hybrid systems [32, §10].

Note. A different approach to the *control* of hybrid systems has been pursued by Kohn and Nerode [114, Appendix II], in which the discrete portion of the dynamics is itself designed as a realizable implementation (that is a sufficient approximation of) some continuous controller. We call this **hybridization**.

We are not really interested in such questions in this thesis. Instead, we wish to view *both plant and controller as hybrid entities*. That is, both are of the form of Figure 1-2. We are motivated to this view by the examples above, such as FVMS. Recall Figure 1-10.

§1.4 HYBRID DYNAMICAL SYSTEMS

DYNAMICAL SYSTEMS

The notion of dynamical system has a long history as an important conceptual tool in science and engineering [6, 67, 73, 96, 117, 133]. It is the foundation of our formulation of hybrid dynamical systems. We review it and some refinements useful in modeling, analysis, and control below, which is condensed from §2.1.2.

Briefly, a **dynamical system** [129] is a system

$$\Sigma = [X, \Gamma, \phi],$$

where X is an arbitrary topological space, the **state space** of Σ. The **transition semigroup** Γ is a topological semigroup with identity. The (**extended**) **transition map** $\phi : X \times \Gamma \to X$ is a continuous function satisfying the identity and semigroup properties [§2.1.2]. A **transition system** is a dynamical system as above, except that ϕ need not be continuous.

Examples of dynamical systems abound, including autonomous ODEs, autonomous difference equations, finite automata, pushdown automata, Turing machines, Petri nets, etc. As seen from these examples, both digital and analog systems can be viewed in this formalism. The utility of this has been noted since the earliest days of control theory [96, 117].

We will also denote by dynamical system the system

$$\Sigma = [X, \Gamma, f],$$

where X and Γ are as above, but the **transition function** f is the **generator** of the extended transition function ϕ.

EXAMPLES. In the case of $\Gamma = \mathbb{Z}$, $f : X \to X$ is given by $f \equiv \phi(\cdot, 1)$. In the case of $\Gamma = \mathbb{R}$, $f : X \to TX$ is given by the vector fields

$$f(x) = \frac{d}{dt} \left. \phi(x,t) \right|_{t=0} .$$

We may also refine the above concept by introducing dynamical systems with initial and final states, input and output, and timing. See §2.1.2.

NOTE. *Timing maps* provide the aforementioned mechanism for reconciling different "time scales," by giving a uniform meaning to different transition semigroups in a hybrid system. This is made clear in §4.

ON TO HYBRID ...

Briefly, a hybrid dynamical system is an indexed collection of dynamical systems along with some map for "jumping" among them (switching dynamical system and/or resetting the state). This jumping occurs whenever the state satisfies certain conditions, given by its membership in a specified subset of the state space. Hence, the entire system can be thought of as a sequential patching together of dynamical systems with initial and final states, the jumps performing a reset to a (generally different) initial state of a (generally different) dynamical system whenever a final state is reached.

More formally, a **general hybrid dynamical system (GHDS)** is a system

$$H = [Q, \Sigma, \mathbf{A}, \mathbf{G}],$$

with its constituent parts defined as follows.

- Q is the set of **index states**, also referred to as **discrete states**.

- $\Sigma = \{\Sigma_q\}_{q \in Q}$ is the collection of **constituent** dynamical systems, where each $\Sigma_q = [X_q, \Gamma_q, \phi_q]$ (or $\Sigma_q = [X_q, \Gamma_q, f_q]$) is a dynamical system as above.

  Here, the X_q are the **continuous state spaces** and ϕ_q (or f_q) are called the **continuous dynamics**.

- $\mathbf{A} = \{A_q\}_{q \in Q}$, $A_q \subset X_q$ for each $q \in Q$, is the collection of **autonomous jump sets**.

- $\mathbf{G} = \{G_q\}_{q \in Q}$, where $G_q : A_q \to \bigcup_{q \in Q} X_q \times \{q\}$, is the collection of (**autonomous**) **jump transition maps**.

  These are also said to represent the **discrete dynamics** of the hybrid dynamical system.

Thus, $S = \bigcup_{q \in Q} X_q \times \{q\}$ is the **hybrid state space** of H. For convenience, we use the following shorthand. $S_q = X_q \times \{q\}$ and $A = \bigcup_{q \in Q} A_q \times \{q\}$ is *the* autonomous jump set. $G : A \to S$ is *the* autonomous jump transition map, constructed componentwise in the obvious way. The **jump destination sets** $\mathbf{D} = \{D_q\}_{q \in Q}$ are given by $D_q = \pi_1[G(A) \cap S_q]$, where π_i is projection onto the ith coordinate. The **switching** or **transition manifolds**,

$M_{q,p} \subset A_q$ are given by $M_{q,p} = G_q^{-1}(p, D_p)$, i.e., the set of states from which transitions from index q to index p can occur.

A GHDS can be pictured as an automaton as in Figure 1-12. There, each node is a constituent dynamical system, with the index the name of the node. Each edge represents a possible transition between constituent systems, labeled by the appropriate condition for the transition's being "enabled" and the update of the continuous state (cf. [69]). The notation ![condition] denotes that the transition *must* be taken when enabled.
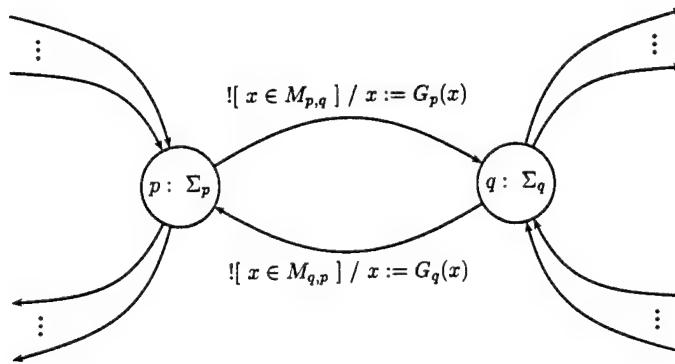
Figure 1-12: Automaton Associated with GHDS.

Roughly,[2] the dynamics of the GHDS H are as follows. The system is assumed to start in some hybrid state in $S \backslash A$, say $s_0 = (x_0, q_0)$. It evolves according to $\phi_{q_0}(x_0, \cdot)$ until the state enters—if ever—A_{q_0} at the point $s_1^- = (x_1^-, q_0)$. At this time it is instantly transferred according to transition map to $G_{q_0}(x_1^-) = (x_1, q_1) \equiv s_1$, from which the process continues. See Figure 1-13.

NOTES.

1. The case $|Q| = 1$ and $\mathbf{A} = \emptyset$ is a single dynamical system.

2. The case $|Q|$ finite, each X_q a subset of $\mathbb{R}^n$, and each $\Gamma_q = \mathbb{R}$ largely corresponds to the *usual* notion of a hybrid system, viz. a coupling of finite automata and differential equations [31, 32, 66]. The two are coupled at "event times" when the continuous state hits certain boundaries, prescribed by the sets A_q.

3. Nondeterminism may be added in the obvious way, i.e., by allowing the possibility that "enabled transitions" need not be taken and letting A be set-valued. Also, see the note below.

4. Other refinements can be made. See §4.1.

In this thesis, a **hybrid dynamical system**, or simply **hybrid system**, is defined as follows:

Definition 1.1 A **hybrid system** *is a general hybrid dynamical system with Q countable, and with $\Gamma_q \equiv \mathbb{R}$ (or $\mathbb{R}_+$) and $X_q \subset \mathbb{R}^{d_q}$, $d_q \in \mathbb{R}_+$, for all $q \in Q$. In the notation above, it*

---

[2]We make more precise statements in §4.3.
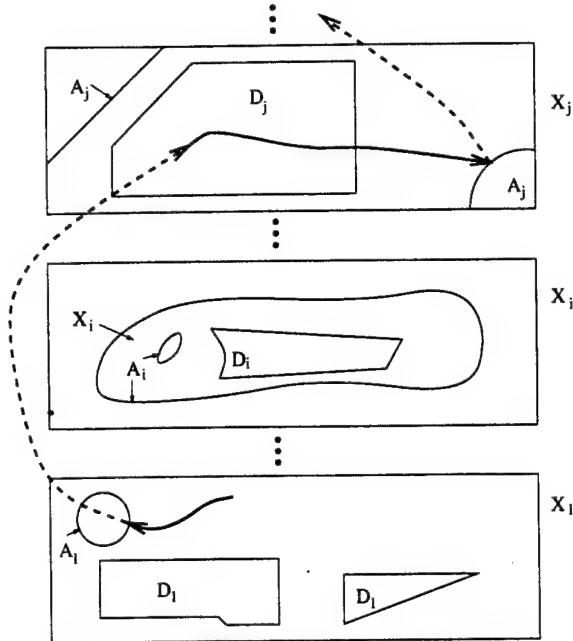
Figure 1-13: Example dynamics of general hybrid dynamical system.

may be written as

$$[Q, [\{X_q\}_{q \in Q}, \mathbb{R}, \{f_q\}_{q \in Q}], \mathbf{A}, \mathbf{G}]$$

where f_q is a vector field on $X_q \subset \mathbb{R}^{d_q}$.

... AND TO HYBRID CONTROL

A controlled general hybrid dynamical system (GCHDS) is a system

$$H_c = [Q, \mathbf{\Sigma}, \mathbf{A}, \mathbf{G}, \mathbf{V}, \mathbf{C}, \mathbf{F}],$$

with its constituent parts defined as follows.

- Q, $\mathbf{A}$, and S are defined as above.

- $\mathbf{\Sigma} = \{\Sigma_q\}_{q \in Q}$ is the collection of controlled dynamical systems, where each $\Sigma_q = [X_q, \Gamma_q, f_q, U_q]$ (or $\Sigma_q = [X_q, \Gamma_q, \phi_q, U_q]$) is a controlled dynamical system as above with (extended) transition map parameterized by **control set** U_q.

- $\mathbf{G} = \{G_q\}_{q \in Q}$, where $G_q : A_q \times V_q \to S$ is the **autonomous jump transition map**, parameterized by the **transition control set** V_q, a subset of the collection $\mathbf{V} = \{V_q\}_{q \in Q}$.

- $\mathbf{C} = \{C_q\}_{q \in Q}$, $C_q \subset X_q$, is the collection of **controlled jump sets**.

- $\mathbf{F} = \{F_q\}_{q \in Q}$, where $F_q : C_q \to 2^S$, is the collection of **controlled jump destination maps**.

As shorthand, G, C, F may be defined as above. Likewise, jump destination sets D_a and D_c may be defined. In this case, $D \equiv D_a \cup D_c$.

Again, a GCHDS has an automaton representation. See Figure 1-14 There, the notation ?[condition] denotes an enabled transition that *may be taken* on command; ":∈" means reassignment to some value in the given set.
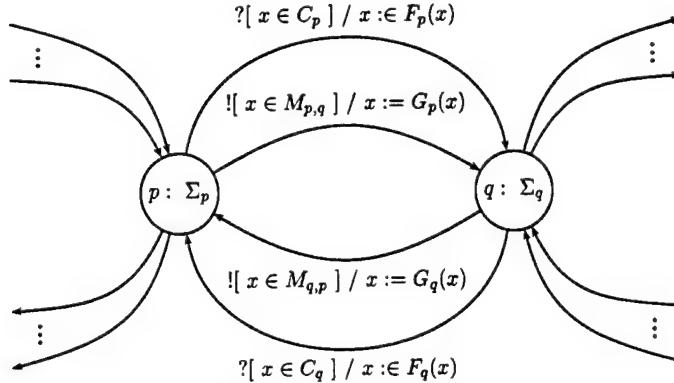
Figure 1-14: Automaton Associated with GCHDS.

Roughly, the dynamics of H_c are as follows. The system is assumed to start in some hybrid state in $S \backslash A$, say $s_0 = (x_0, q_0)$. It evolves according to $\phi_{q_0}(\cdot, \cdot, u)$ until the state enters—if ever—either A_{q_0} or C_{q_0} at the point $s_1^- = (x_1^-, q_0)$. If it enters A_{q_0}, then it *must* be transferred according to transition map $G_{q_0}(x_1^-, v)$ for some chosen $v \in V_{q_0}$. If it enters C_{q_0}, then we *may* choose to jump and, if so, we may choose the destination to be any point in $F_{q_0}(x_1^-)$. In either case, we arrive at a point $s_1 = (x_1, q_1)$ from which the process continues. See Figure 1-15.

> NOTE. **Nondeterminism** in transitions may be taken care of by partitioning ?[condition] into those which are controlled and uncontrolled (cf. [74]). **Disturbances** (and other nondeterminism) may be modeled by partitioning U, V, and C into portions that are under the influence of the controller or nature respectively. Systems with state-output, edge-output, and autonomous and controlled jump delay maps (Δ_a and Δ_c, respectively) may be added as above. See §4.1 for more details.

Our "unified" model for hybrid control is detailed in §5. Briefly, it is a *controlled hybrid system*, with the form

$$\left[\mathbb{Z}_+, \left[\{\mathbb{R}^{d_i}\}_{i=0}^{\infty}, \mathbb{R}_+, \{f_i\}_{i=0}^{\infty}, \mathbf{U} \right], \mathbf{A}, \mathbf{V}, \mathbf{G}, \mathbf{C}, \mathbf{F} \right],$$

where each $d_i \in \mathbb{Z}_+$. In §5 we show that this model encompasses the discrete phenomena associated with hybrid systems [§3.2] as well as subsumes previously posed hybrid systems models [5, 9, 38, 114, 135, 152, §§3.3–3.8].
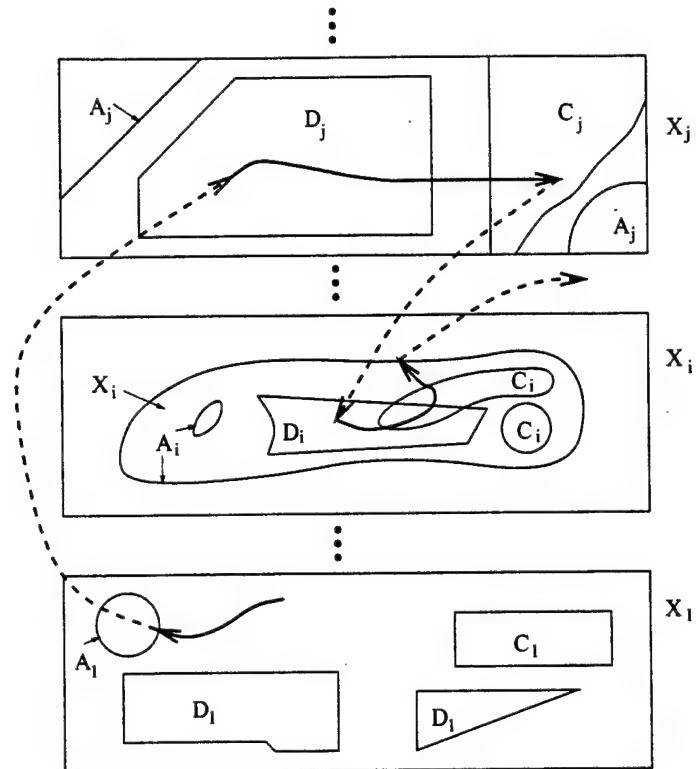
Figure 1-15: Example dynamics of general controlled hybrid dynamical system.

§1.5 MODELING CONTRIBUTIONS

REVIEW

Evidently, a hybrid system has continuous dynamics modeled by a differential equation

$$\dot{x}(t) = \xi(t), \qquad t \geq 0$$

that depends on some discrete phenomena. Here, $x(t)$ is the *continuous component* of the state taking values in some subset of a Euclidean space. $\xi(t)$ is a *controlled vector field* that generally depends on $x(t)$, the *continuous component* $u(t)$ of the control policy, and the aforementioned discrete phenomena.

An examination of real-world examples and a review of other hybrid systems models has led us to an identification of these phenomena. The discrete phenomena generally considered are as follows.

1. **Autonomous switching:** Here the vector field $\xi(\cdot)$ changes discontinuously when the state $x(\cdot)$ hits certain "boundaries."

2. **Autonomous jumps:** Here the continuous state $x(\cdot)$ jumps discontinuously on hitting prescribed regions of the state space.

3. **Controlled switching:** Here the vector field $\xi(\cdot)$ changes abruptly in response to a control command.

4. **Controlled jumps**: Here the continuous state $x(\cdot)$ changes discontinuously in response to a control command.

We also review in some detail the hybrid systems models of Witsenhausen (WHS), Tavernini (TDA), Back-Guckenheimer-Meyers (BGM), Nerode-Kohn (NKSD), Antsaklis-Stiver-Lemmon (ASL), and Brockett (BB/BD/BDV). We summarize this review as follows.

The WHS, TDA, NKSD, and ASL models combine ordinary differential equations with finite (or digital) automata by allowing the ODEs to depend on the automaton's state or output, while the automaton's state or input depends on a partitioning of the continuous state space. The discrete state/input change on crossing these partition boundaries. Each model places different restrictions on the allowed partitioning. All these models implement autonomous switching.

The BGM model is similar, except that it allows one to make autonomous jumps in the continuous state, set parameters, or start timers upon hitting the partition boundaries. It uses autonomous switching and autonomous jumps.

Brockett combines differential equations with a single rate equation, or "clock," which progresses monotonically. The differential equations depend on the integer value of this clock variable. He also combines finite automata that update on the times when the clock passes through integer values. His BD model is a special form of autonomous jumps.

Comparing autonomous versions of each model, we see that BGM *contains* each of the others, while TDA is contained in each of the other autonomous-switching models. However, BD and the autonomous-switching models are not strictly comparable.

BGM and TDA are only autonomous. From the control perspective, NKSD and ASL models focus on the "control automaton," coding the action of the controller in the mappings from continuous states to input symbols, through automaton to output symbols, and back to controls. See Figure 1-1.

Witsenhausen adds a control to the continuous component of the system dynamics. Brockett's BD/BDV models allow the possibility of both continuous and discrete controls to be exercised as input to the continuous and symbolic dynamics of the systems, respectively. These last are closer to our own study of hybrid control below. See Figure 1-2.

CLASSIFICATION

In this chapter we classify hybrid systems according to their structure and the phenomena that they exhibit. The hierarchy of classes we explore are as follows (for both autonomous and controlled systems). First, there are *general hybrid dynamical systems* (*GHDS*). These are then refined to the concept of *hybrid dynamical system*, or simply *hybrid system*, studied in this thesis. Then there are two restrictions of hybrid systems, in which the discrete dynamics are suppressed, called *switched systems* and *continuous switched systems*, which are analyzed in §§8–9. A further taxonomy of such systems in terms of their structure and the discrete phenomena which they admit is presented.

Along the way we also discuss the dynamics of general hybrid dynamical systems. In particular, we give sufficient conditions on GHDS which allow us to construct its **positive orbit**, i.e., the set of points reachable in forward "time flow."

Let $I \equiv S \backslash A$. When a GHDS is time-uniform (all semigroups identical) with time-like semigroup Γ, it induces a Γ^+-transition system $[I, \Gamma^+, \Phi]$. In this case, we may define its (forward) **trajectory** as a *function* from Γ^+ into I. We give sufficient conditions for existence and uniqueness of these trajectories.

Finally, we also introduce *switched systems*, for example,

$$\dot{x}(t) = f_i(x(t)), \qquad i \in Q \simeq \{1, \ldots, N\},$$

where $x(t) \in \mathbb{R}^n$ and Q is called the *switching set*. We add the following *switching rules*.

- Each f_i is globally Lipschitz continuous.

- The i's are picked in such a way that there are finite switches in finite time.

We also consider discrete-time versions. Abstracting away of the finite dynamics in studying switched systems above can be motivated by "verification by successive approximation" [2].

A *continuous* switched system is one whose vector fields agree at switching times.

UNIFIED MODEL

Finally, we come to our unified model for hybrid control. We consider a controlled hybrid systems model, i.e.,

$$\left[\mathbb{Z}_+, \left[\{\mathbb{R}^{d_i}\}_{i=0}^\infty, \mathbb{R}_+, \{f_i\}_{i=0}^\infty, U \right], \mathbf{A}, \mathbf{G}, V, \mathbf{C}, \mathbf{D} \right].$$

To ease presentation, but without real loss of generality, we consider the continuous and discrete control sets to be uniform, and that the controlled jump destinations are given by the sets $D_i \in \mathbf{D}$ instead of by the set-valued maps F_i.

We also add delay operators on autonomous and controlled jumps:

- **autonomous jump delay** $\Delta_a : A \times V \to \mathbb{R}_+$.

- **controlled jump delay** $\Delta_c : C \times D_c \to \mathbb{R}_+$.

The dynamics of the control system is much the same as for GCHDS above, except that the delay maps give rise to a sequence of **pre-jump times** $\{\tau_i\}$ and another sequence of **post-jump times** $\{\Gamma_i\}$ satisfying $0 = \Gamma_0 \leq \tau_1 < \Gamma_1 < \tau_2 < \Gamma_2 < \cdots \leq \infty$. On each interval $[\Gamma_{j-1}, \tau_j)$ with non-empty interior, $x(\cdot)$ evolves according to $\dot{x}(t) = f_i(x(t), u(t))$ in some X_i, $i \in \mathbb{Z}_+$. At the next pre-jump time (say, τ_j) it jumps to some $D_k \in X_k$ according to one of the following two possibilities:

1. $x(\tau_j) \in A_i$, in which case it *must* jump to $x(\Gamma_j) = G_i(x(\tau_j), v_j) \in D$ at time $\Gamma_j = \tau_j + \Delta_{a,i}(x(\tau_j), v_j)$, $v_j \in V$ being a control input. We call this phenomenon an autonomous jump.

2. $x(\tau_j) \in C_i$ and the controller *chooses* to—it does not have to—move the trajectory discontinuously to $x(\Gamma_j) \in D$ at time $\Gamma_j = \tau_j + \Delta_{c,i}(x(\tau_j), x(\Gamma_j))$. We call this a controlled (or impulsive) jump.

See Figure 1-15.

Thus, the **admissible control actions** available are

- the **continous controls** $u(\cdot)$, exercised in each constituent regime,

- the **discrete controls** $\{v_i\}$ exercised at the pre-jump times of autonomous jumps (which occur on hitting the set A),

- the pre-jump or **intervention times** $\{\zeta_i\}$ and associated **destinations** $\{x(\zeta_i')\}$ of the controlled jumps.

We then explicitly show that the above model captures all identified discrete phenomena arising in hybrid systems and subsumes all reviewed and classified hybrid systems models. The resulting model is useful for posing and solving hybrid control problems in the sequel.

§1.6 ANALYSIS CONTRIBUTIONS

TOPOLOGICAL RESULTS

In traditional feedback control systems—continuous-time, discrete-time, sampled-data—the maps from output measurements to control inputs are continuous (in the usual metric-based topologies). Continuity of state evolution and controls with respect to the states also plays a role. Yet, in general, hybrid systems are not continuous in the initial condition:

Example 1.2 *Consider the following hybrid system on $X_1 = X_2 = \mathbb{R}^2$. The continuous dynamics is given by $f_1 \equiv (1,\ 0)^T$ and $f_2 \equiv (0,\ 1)^T$. The discrete dynamics is given by $A_1 = [0,\ 1]^2$ and $G(x, 1) = (x, 2)$. Now consider the initial conditions $x(0) = (-\epsilon, -\epsilon)^T$ and $y(0) = (-\epsilon, 0)^T$. Note that $x(1) = (1 - \epsilon, -\epsilon)$ but $y(1) = (0, 1 - \epsilon)$. Clearly, no matter how small ϵ, hence $\|x(0) - y(0)\|_\infty$, is chosen, $\|x(1) - y(1)\|_\infty = 1$.*

We examine systems as in Figure 1-1, where the set of symbols, automaton states, and outputs, are finite sets and the plant and controls belong to a continuum. However, note that the only continuous maps from a connected set to a disconnected one are the constant ones. Hence, the usual discrete topologies on a set of symbols do not lead to nontrivial continuous AD maps.

We then examine topologies that lead to continuity of each member of topologies for which the AD maps from measurements to symbols *are* continuous. In dynamics terms, we examine topologies that lead to continuity of each member of the family of maps $G \circ \phi_q$. One topology in particular, proposed by Nerode and Kohn [114], is studied in depth.

We then look at what happens if we attempt to "complete the loop" in Figure 1-1, by also considering the DA maps. We exhibit a topology making the whole loop continuous. But, instead of dwelling on this, we examine a different view of hybrid systems as a set of continuous controllers, with switching among them governed by the discrete state. Also, we examine fuzzy control systems consisting of a finite set of so-called fuzzy rules. On the surface, they are hybrid. Yet, we show that fuzzy control leads to continuous maps (from measurements to controls) and that all such continuous maps may be implemented via fuzzy control.   -

COMPLEXITY RESULTS

Computational equivalence (or simulation of computational capabilities) may be shown in the following two ways:

1. Comparing accepted languages [76],

2. Simulation of step-by-step behavior [20].

The above are clear when comparing two digital automata. The situation is slightly harder in comparing digital and analog systems.

In order to examine the computational capabilities of hybrid and continuous systems we first must introduce notions of a continuous-time system simulating a discrete-time one. [31, §7.2]:

Definition 1.3 *A continuous-time transition system* $[X, \mathbb{R}_+, f]$ **simulates via section** *or* **S-simulates** *a discrete-time transition system* $[Y, \mathbb{Z}_+, F]$ *if there exist a continuous surjective partial function* $\psi : X \rightarrow Y$ *and* $t_0 \in \mathbb{R}_+$ *such that for all* $x \in \psi^{-1}(Y)$ *and all* $k \in \mathbb{Z}_+$

$$\psi(f(x, kt_0)) = F(\psi(x), k).$$

Definition 1.4 *A continuous-time transition system* $[X, \mathbb{R}_+, f]$ **simulates via intervals** *or* **I-simulates** *a discrete-time transition system* $[Y, \mathbb{Z}_+, F]$ *if there exist a continuous surjective partial function* $\psi : X \rightarrow Y$ *and* $\epsilon > 0$ *such that* $V \equiv \psi^{-1}(Y)$ *is open and for all* $x \in V$ *the set* $T = \{t \in \mathbb{R}_+ \mid f(x, t) \in V\}$ *is a union of intervals* (τ_k, τ'_k), $0 = \tau_0 < \tau'_0 < \tau_1 < \tau'_1 < \cdots, |\tau'_k - \tau_k| \geq \epsilon$, *with*

$$\psi(f(x, t_k)) = F(\psi(x), k),$$

for all $t_k \in (\tau_k, \tau'_k)$.

When the continuous-time transition system is a HDS, the maps ψ above can be viewed as an edge-output and state-output map, respectively. In this case, S-simulation can be viewed as equivalent behavior at equally-spaced edges and I-simulation as equivalent behavior in designated nodes. S- and I-simulation are distinct notions. SI-simulation denotes the case when both hold.

In §7 we show the following:

- Every dynamical system $[\mathbb{R}^n, \mathbb{Z}, F]$ can be S-simulated by an autonomous-switching, two-discrete-state hybrid system on $\mathbb{R}^{2n}$.

- Every dynamical system $[\mathbb{R}^n, \mathbb{Z}_+, F]$ can be S-simulated by an autonomous-jump, two-discrete-state hybrid system on $\mathbb{R}^n$.

- Every dynamical system $[Y, \mathbb{Z}, F]$, $Y \subset \mathbb{Z}^n$, can be SI-simulated by a (continuous) dynamical system of the form $[\mathbb{R}^{2n+1}, \mathbb{R}_+, f]$. Furthermore, if Y is bounded f can be taken Lipschitz continuous.

As corollaries to the last we result, we have (via demonstrated isomorphisms with dynamical systems on $\mathbb{Z}$)

- Every Turing machine, pushdown automata, and finite automaton can be SI-simulated by a (continuous) dynamical system of the form $[\mathbb{R}^3, \mathbb{R}_+, f]$.

- Using SI-simulation, there is a system of continuous ODEs in $\mathbb{R}^3$ with the power of universal computation.

Noting that even ordinary dynamical systems are so computationally powerful, we use the the famous asynchronous arbiter problem [26, 100, 144] to distinguish between dynamical and hybrid systems.

NOTE. An **arbiter** is a device that can be used to decide the winner of two-person races (within some tolerances). It has two input buttons, B_1 and B_2, and two output lines, W_1 and W_2, that can each be either 0 or 1. For its "technical specifications" see §7.4.1.

In particular, we settle the problem in an ODE framework by showing that no system of the form

$$[\mathbb{R}^n, \mathbb{R}_+, B, f, W, h],$$

with f Lipschitz and h continuous, can implement an arbiter [26, 31]. On the other hand, we exhibit a hybrid system of the form

$$[\{1, 2\}, [\mathbb{R}^n, \mathbb{R}_+, B, \{f_1, f_2\}, W, h], \mathbf{A}, \mathbf{G}],$$

with each f_q Lipschitz, h continuous, and $\mathbf{G}$ autonomous-switching, that satisfies the arbiter specifications.

ANALYSIS TOOLS

In the first part of §8, we develop general tools for analyzing continuous switching systems. For instance, we prove an extension of Bendixson's Theorem to the case of Lipschitz continuous vector fields. This gives us a tool for analyzing the existence of limit cycles of continuous switching systems. We also prove a lemma dealing with the continuity of differential equations with respect to perturbations that preserve a linear part. Colloquially, this lemma demonstrates the robustness of ODEs with a linear part. For purpose of discussion, we call it the *Linear Robustness Lemma*. This lemma is useful in easily deriving some of the common robustness results of nonlinear ODE theory (as given in, for instance, [11]). This lemma also becomes useful in studying singular perturbations if the fast dynamics are such that they maintain the corresponding algebraic equation to within a small deviation. We add some simple propositions that allow us to do this type of analysis in §9.

In the second part of §8, we examine stability of we introduce "multiple Lyapunov functions" as a tool for analyzing Lyapunov stability of switched systems. The idea here is to impose conditions on switching that guarantee stability when we have Lyapunov functions for each system f_i individually. Iterative function systems are presented as a tool for proving Lagrange stability and positive invariance. We also address the case where the finite switching set is replaced by an arbitrary compact set.

ANALYZING EXAMPLES

We have analyzed example systems arising from a realistic aircraft controller problem which logically switches between two controllers (one for tracking and one for regulation about a fixed angle of attack) in order to achieve reasonable performance and safety. While stability of such hybrid systems has previously only been examined using simulation [134], we were able to prove global asymptotic stability for a meaningful class of cases [28]. Using our robustness lemma to compare ODE solutions, we extend the result to a class of *continuations* of these systems, in which dynamically smooths the logical nonlinearity.

The conclusion in this case is that the continuation method worked in reverse, i.e., it was easier to prove stability of the original, hybrid system directly. Furthermore, we concluded stability of the continuation via that of the original system. In effect, we showed robustness of the max system to the considered class of dynamic continuations.

§1.7 CONTROL CONTRIBUTIONS

THEORETICAL RESULTS

We consider the following optimal control problem on our unified hybrid systems model.

Let $a > 0$ be a **discount factor**. We add to our model the following known maps:

- **Running cost** $k : S \times U \to \mathbb{R}_+$.

- **Autonomous jump cost** or **transition cost** $c_a : A \times V \to \mathbb{R}_+$.

- **Controlled jump cost** or **impulse cost** $c_c : C \times D_c \to \mathbb{R}_+$.

The total discounted cost is defined as

$$\int_{\mathbb{T}} e^{-at} k(x(t), u(t)) \, dt + \sum_i e^{-a\sigma_i} c_a(x(\sigma_i), v_i) + \sum_i e^{-a\zeta_i} c_c(x(\zeta_i), x(\zeta_i')) \qquad (1.2)$$

where $\mathbb{T} = \mathbb{R}_+ \setminus (\bigcup_i [\tau_i, \Gamma_i))$, $\{\sigma_i\}$ (respectively $\{\zeta_i\}$) are the successive pre-jump times for autonomous (respectively impulsive) jumps and ζ_j' is the post-jump time for the jth impulsive jump. The **decision** or **control** variables over which Equation (1.2) is to be minimized are the *admissible controls* of our unified model.

Under some assumptions (the necessity of which are shown via examples) we have the following results [32]:

- A finite optimal cost exists for any initial condition. Furthermore, there are only finitely many autonomous jumps in finite time.

- Using the relaxed control framework, an optimal trajectory exists for any initial condition.

- For every $\epsilon > 0$ an ϵ-optimal control policy exists wherein $u(\cdot)$ is precise, i.e., a Dirac measure.

- The value function, V, associated with the optimal control problem is continuous on $S \setminus (\partial A \cup \partial C)$ and satisfies the following **generalized quasi-variational inequalities (GQVIs)**.

  1. $x \in S \setminus A$:
  $$F(x, u) \equiv \langle \nabla_x V(x), f_i(x, u) \rangle - aV(x) + k(x, u),$$
  $$\min_u F(x, u) \leq 0.$$

  2. On C:
  $$V(x) \leq \min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x,z)} V(z) \right\}$$

  3. On A:
  $$V(x) \leq \min_v \left\{ c_a(x, v) + e^{-a\Delta_a(x,v)} V(G(x, v)) \right\}$$

  4. On C:
  $$(1) \cdot (2) = 0$$

ALGORITHMS AND EXAMPLES

In this section we outline four approaches to solving the generalized quasi-variational inequalities (GQVIs) associated with optimal hybrid control problems. Our algorithmic basis for solving these GQVIs is the Bellman Equation:

$$V^*(x) = \min_{p \in \Pi} \left\{ g(x, a) + V^*(x'(x, a)) \right\},$$

where Π is a generalized set of actions. The three classes of actions available in our hybrid systems framework at each x are

- **Continuous Controls:** $u \in U$.

- **Controlled Jumps:** choosing source and destination (if $x \in C$).

- **Autonomous Jumps:** possibly modulated by discrete controls $v \in V$ (if $x \in A$).

From this viewpoint, generalized policy and value iteration become solution tools.

The key to *efficient* algorithms for solving optimal control problems for hybrid systems lies in noticing their strong connection to the models of impulse control [§2.2.4] and piecewise-deterministic processes [§2.2.5]. Making this explicit, we develop algorithms similar to those for impulse control and one based on linear programming.

Three illustrative examples are solved. They are as follows. First, we consider a hysteresis system that exhibits autonomous switching and has a continuous control. Then we discuss a satellite station-keeping problem. The on-off nature of the satellite's reaction jets creates a system involving controlled switching. We end with a transmission problem. The goal is to find the *hybrid* strategy of continuous accelerator input and discrete gear-shift position to achieve maximum acceleration. In each case, the optimal controls produced verify engineering intuition.

Chapter 2

Preliminaries and Related Work

In this chapter, we cover preliminaries and review some literature related to hybrid systems.

§2.1 PRELIMINARIES

Throughout the thesis we assume familiarity with the notation and concepts of analysis [126], topology [62, 75, 113, §A], ordinary differential equations [6, 73], automata theory [20, 76], control theory [58, 85, 96, 133], and nonlinear systems analysis [132, 142].

Next, we collect some notation used throughout. Then we cover the necessary preliminary information from dynamical systems, ordinary differential equations, and automata theory. Some review of topology is done in §A. A majority of the notation is collected in the Symbol Index, pp. 193–194. Finally, a general Index is provided for convenience in locating definitions.

§2.1.1 NOTATION

New concepts being defined are displayed in **bold face**. The end of a proof is denoted with the symbol $\square$. Sectional cross-referencing within the thesis is done using the symbol §.

A **system** is the abstract entity of our study that we shall not formally define, just as "points" are not defined in analysis. Difference and differential equations are examples of frequently used systems. A **map** is a function; we use the two interchangeably.

We make use of common abbreviations like ODEs (ordinary differential equations), FA (finite automata/on), DEDS (discrete event dynamical systems; see [74]), etc.

The symbols $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$, and $\mathbb{Z}_+$ denote the reals, nonnegative reals, integers, and nonnegative integers, respectively. For $x \in \mathbb{R}$, $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x, and, in an abuse of common notation, $\lceil x \rceil$ denotes the least integer *greater than* x. $|A|$, A a set, denotes its cardinality; $A \simeq B$, when A and B are sets, is read A is **isomorphic** to B and means herein that $|A| = |B|$; $\underline{N}$ denotes the set $\{1, 2, \ldots, N\}$; $\equiv$ means "defined equal to," whereas := denotes reassignment of meaning.

Below we deal with continuous time systems, such as ODEs, that are affected by events at discrete instants, such as jumps in state. We use $[t]$ to denote the time less than or equal to t at which the last "jump" or "event" occurred. If the event is unclear, we subscript the variable, so that $[t]_p$ denotes the time at which the variable p last jumped. Throughout, $v\lfloor p \rfloor$ and $x[t]$ are shorthand for $v(\lfloor p \rfloor)$ and $x([t])$, respectively (cf. [38]). Finally, we use Sontag's discrete-time shift operator [133] to denote the discrete phenomena updating a hybrid systems' states. That is, q^+ is the successor of q. Likewise, q^- is its predecessor.

Other notation is common [113, 126]. For example, $X \backslash U$ represents the complement

of U in X; $\overline{U}$ represents the closure of U, U^o its interior, ∂U its boundary; $f(t^+), f(t^-)$ denote the right-hand and left-hand limits of the function f at t, respectively; a function is right-continuous if $f(t^+) = f(t)$ for all t; $C(X, Y)$ denotes the space of continuous functions with domain X and range Y; v^T denotes the transpose of vector v; unless specified $\|x\|$ denotes an arbitrary norm of vector x, $\|x\|_2$ its Euclidean norm; the infinity norm of $x \in \mathbb{R}^n$, denoted $\|x\|_\infty$, is $\max_{i=1}^n |x_i|$.

§2.1.2 DYNAMICAL SYSTEMS

First, we review some standard definitions from dynamical systems [67, 129]. A **dynamical system** is a system $\Sigma = [X, \Gamma, \phi]$, where

- X is an arbitrary topological space, the **state space** of Σ.

- The **transition semigroup** Γ is a topological semigroup.[1]

- The (**extended**) **transition map** $\phi : X \times \Gamma \to X$ satisfies the following.

   1. **Identity.** For all $x \in X$, $\phi(x, 0) = x$.
   2. **Semigroup.** For all $x \in X$ and arbitrary g_1, g_2 in Γ,

$$\phi(\phi(x, g_1), g_2) = \phi(x, g_1 + g_2).$$

   3. **Continuity.** ϕ is continuous in both arguments simultaneously, i.e., for any neighborhood W of the point $\phi(x, g)$ there exist neighborhoods U and V of the point x and the element g respectively such that $\phi(U, V) \subset W$.

Such objects are well-studied in mathematics under the names *topological transformation groups* or *continuous general dynamical system*. A **general dynamical system** need not have Γ a topological space and only requires continuity with respect to x [129].

Technically, we have defined **semi-dynamical systems**, with the term dynamical system reserved for the case Γ above is a group. However, the more "popular" notion of dynamical system in math and engineering—and the one used here—requires only the semigroup property [59, 96]. Thus, the term **reversible** dynamical system is used when it is necessary to distinguish the group from semigroup case [97].

If a dynamical system is defined on a subset of X, we say it is a **dynamical system in** X. For every fixed value of the parameter g, the function $\phi(\cdot, g)$ defines a mapping of the space X into itself. Given $[X, \mathbb{Z}_+, \phi]$, $\phi(\cdot, 1)$ is its **transition function**. Thus if $[X, \mathbb{Z}_+, \phi]$ is reversible, its transition function is invertible, with inverse given by $\phi(\cdot, -1)$.

Generalizing this notion, we *also denote by dynamical system*[2] a system

$$\Sigma = [X, \Gamma, f],$$

where X and Γ are as above, but the **transition function** f is the **generator** of the *extended* transition function ϕ. Thus $f : X \times \Gamma_g \to X$, where Γ_g generates Γ. For example,

---

[1] We assume only semigroups with identity, (a.k.a. **monoids**). We use addition notation for the semigroup operation throughout (i.e., $+$ is the operation, 0 the identity) and suppress the operation in referencing the semigroup.

[2] The distinction should be clear by context.

in the case of $\Gamma = \mathbb{Z}$, $f : X \to X$ is given by $f \equiv \phi(\cdot, 1)$. In the case of $\Gamma = \mathbb{R}$, $f : X \to TX$ is given by

$$f(x) = \frac{d}{dt} \phi(x, t)\Big|_{t=0}.$$

This definition is just as general as before since f may depend on all of Γ. Of particular interest are cases where Γ is finitely generated. For example, we had it depend on 1 (the generator for $\mathbb{Z}$) and dt (informally, the generator for $\mathbb{R}$) in the examples above. Another concrete example is a finite automaton given by its transition function, $\nu : Q \times I \to Q$ (since I generates the free monoid I^* on which its extended transition function is defined) [20].

Examples of dynamical systems abound, including autonomous ODEs, autonomous difference equations, finite automata, pushdown automata, Turing machines, Petri nets, etc.

The *set* $\phi(x, \Gamma) = \{\phi(x, g) : g \in \Gamma\}$ is called the **orbit** of the point x. The *function* $\Gamma \to \phi(x, \Gamma)$ **trajectory** of the point x. An **equilibrium** or **fixed point** of $[X, \Gamma, \phi]$ is a point x such that $\phi(x, g) = x$ for all $g \in \Gamma$. A set $A \subset X$ is **invariant** with respect to ϕ, if $\phi(A, g) \subset A$ for all $g \in \Gamma$.

The notions of equivalence and homomorphism are crucial. Two dynamical systems $[X, \Gamma, \phi]$, $[Y, \Gamma, \theta]$ are said to be **isomorphic** or (**topologically**) **equivalent** if there exists a homeomorphism $\psi : X \to Y$ such that

$$\psi(\phi(x, g)) = \theta(\psi(x), g),$$

for all $x \in X$ and $g \in \Gamma$. If the mapping ψ is only continuous, then $[X, \Gamma, \phi]$ is said to be **homomorphic** to $[Y, \Gamma, \theta]$. Homomorphisms preserve trajectories, fixed points, and invariant sets.

Less restrictively, a **transition system** is a *general* dynamical system as above, except that ϕ need not be continuous in x. As shorthand we use **Γ-transition system** to denote one with group Γ. **Continuous-** and **discrete-time transition system** denote the cases where $\Gamma = \mathbb{R}$ (or $\mathbb{R}_+$) and $\Gamma = \mathbb{Z}$ (or $\mathbb{Z}_+$), respectively.

Next, we refine the above concept by introducing controlled dynamical systems and systems with marked states, output, and timing. In each case, X, Γ, ϕ, and f are as above.

A **controlled dynamical system** or a dynamical system **with input** is a system

$$\Sigma = [X, \Gamma, U, f], \qquad (\text{or } \Sigma = [X, \Gamma, U, \phi]),$$

where U is the set of inputs. The input $u \in U$ may act state-by-state, i.e. $f : X \times \Gamma_g \times U \to X$, or by transition, i.e. $\phi : X \times \Gamma \times U \to X$.

To each of the above, we may append the following refinements:

- **Output.** Add Y the set of outputs. The output map h may produce **state-output**, i.e. $h : X \to Y$, or **edge-output**, i.e. $h : X \times \Gamma \to Y$.

  NOTE. These are often equivalent. For example, **Moore** (resp. **Mealy**) **machines** are finite automata with state-output (resp. edge-output). However, Moore and Mealy machines are equivalent [76]. In differential equations one almost always assumes state-output, the edge-output case being subsumed in most cases of interest by integration operators. There is also the equivalence of nodes and actions under perfect recall in game theory [115].

- **Marked states.** These are distinguished subsets of X. For example, let $I \subset X$ denote a set of (**admissible**) **initial** or **start states**, and $F \subset S$ the set of **final**, **halting**, or **cemetery states**. Roughly, such a dynamical system, is defined to start from any point I, from which it evolves until it hits a point in F, at which time its action is halted.

- **Timing.** Add a **transition time map** or **timing map**. $\tau : X \times \Gamma \to \mathbb{R}$ (or $\mathbb{R}_+$). Of particular interest are maps where τ is constant on X and those that, in addition, are homomorphisms with subsets of $\mathbb{R}$ ($\mathbb{R}_+$). In the generator case, the timing map can be defined on $X \times \Gamma_g$.

§2.1.3 ODEs

In this thesis, the continuous dynamical systems dealt with are defined by the solutions of **ordinary differential equations (ODEs)** [73]:

$$\dot{x}(t) = f(x(t)), \tag{2.1}$$

where $x(t) \in X \subset \mathbb{R}^n$. The function $f : X \to \mathbb{R}^n$ is called a **vector field** on $\mathbb{R}^n$. The resulting dynamical system is then given by $\phi(x_0, t) = x(t)$ where $x(\cdot)$ is the solution to Equation (2.1) starting at x_0 at $t = 0$. We assume existence and uniqueness of solutions; see [73] for conditions. A well-known *sufficient* condition is that the vector field f is **Lipschitz continuous**. That is, there exists $L > 0$ (called the **Lipschitz constant**) such that

$$\|f(x) - f(y)\| \leq L\|x - y\|, \qquad \text{for all } x, y \in X.$$

A system of ODEs is called **autonomous** or **time-invariant** if its vector field does not depend explicitly on time. Throughout, the shorthand **continuous** (resp. **Lipschitz**) **ODEs** to denote those with continuous (resp. Lipschitz) vector fields.

An ODE **with inputs and outputs** [96, 133] or **plant** is given by

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \\ y(t) &= h(x(t)), \end{aligned} \tag{2.2}$$

where $x(t) \in X \subset \mathbb{R}^n$, $u(t) \in U \subset \mathbb{R}^m$, $y \in Y \subset \mathbb{R}^p$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and $h : \mathbb{R}^n \to \mathbb{R}^p$. The functions $u(\cdot)$ and $y(\cdot)$ are the **inputs** and **outputs**, respectively.

Whenever inputs are present, as in Equation (2.2), we say f is a **controlled vector field**.

§2.1.4 DIGITAL AUTOMATA

We begin with some standard material [76]. A **symbol** is the abstract entity of automata theory. Examples are letters and digits. An **alphabet** is a finite set of symbols. A **string** is a finite sequence of juxtaposed symbols. The **empty string**, denoted ε, is the string consisting of zero symbols.

A **finite automaton (FA)** is a system (Q, I, ν, q_0, M), where

- Q is a finite set of **states**,

- I is an alphabet, called the **input alphabet**,

- ν is the **transition function** mapping $Q \times I$ into Q,

- $q_0 \in Q$ is the **initial state**,

- $M \subset Q$ is the set of the **accepting states**.

We envision a finite automaton as a finite control reading a sequence of symbols from I written on a tape. In one move, the finite automaton in state q scans symbol a, enters state $\nu(q, a)$, and moves its head one symbol to the right. If the head moves off the right end of the tape while in an accepting state, then it accepts the entire tape.

A (**deterministic**) **pushdown automaton** (**in normal form**), or simply **PDA**, is a system $M = (Q, \Gamma, I, \nu, q_0, Z_0, F)$, where

- Q is the finite set of **states**,

- Γ is an alphabet, called the **tape alphabet**,

- I is an alphabet, called the **input alphabet**,

- ν is the **transition function** from $Q \times \Gamma \times (I \cup \{\varepsilon\})$ to $Q \times (\Gamma \cup \{\varepsilon\})$,

- q_0 in Q is the **start state**,

- Z_0 in Γ is the **start symbol**,

- $F \subset Q$ is the set of **final states**.

Thus, we envision a finite control reading a sequence of symbols of I written on a tape and manipulating a stack of symbols. In state q and with Z the topmost stack symbol, then exactly one of the following is true.[3]

1. $\nu(q, Z, i) = (p, Y)$ for some $i \in I$,

2. $\nu(q, Z, \varepsilon) = (p, Y)$.

If $Y \in \Gamma$, the PDA performs a **push** operation, i.e., *it replaces Z with ZY, and Y becomes the new top stack symbol*. If $Y = \varepsilon$, it performs a **pop**, i.e., *it removes Z from the stack*. In the first case, the PDA performs a push or pop (depending on Y), moves to state p, and advances the input head. In the second case, the PDA performs a push or pop (depending on Y) and moves to state p (i.e., it ignores the tape input and does not advances the input head). The PDA is assumed to halt on an empty stack or the state's entering F.

In the 1930's Alan Turing introduced a mechanical model for computation which consisted of a finite control, an (infinite to the right) input tape that is divided into cells, and a tape head that scans one cell of the tape at a time. Each tape cell may hold one of a finite number of tape symbols. Initially, the machine starts with its tape head at the leftmost cell, its internal control at some designated initial state, and its input coded into the first $0 \le n < \infty$ tape cells. All remaining cells hold a special symbol, called the blank, which is separate from the input symbols. More formally a **Turing machine** (**TM**) is a system

$$M = (Q, \Gamma, B, I, \nu, q_0, F),$$

where

---

[3]By determinism. Also we may assume ν is a total function by appending special final states.

- Q is the finite set of **states**,

- Γ an alphabet of allowable **tape symbols**,

- B, a symbol of Γ, is the **blank**,

- I, a subset of Γ not including B, is the alphabet of **input symbols**,

- ν is the **next move function**, a mapping from $Q \times \Gamma$ to $Q \times \Gamma \times \{L,\ R\}$,

- q_0 in Q is the **start state**,

- $F \subset Q$ is the set of **final states**.

Evidently, a Turing machine is a finite automaton whose input at step k is the tape symbol in the cell over which the tape head is positioned at the kth step. In addition to changing state, however, the Turing machine can also print a symbol on the scanned tape cell and move its head left or right one cell.

Now we move on to some particular notation. An **inputless FA** (resp. **PDA**) is one whose input alphabet is empty, i.e., one whose transition function depends solely on its state (resp. state and top stack symbol).

A **digital** or **symbolic automaton** (with input and output) is a system (Q, I, ν, O, η), consisting of the state space, input alphabet, transition function, output alphabet, and output function, respectively. We assume that Q, I, and O are each isomorphic to subsets of $\mathbb{Z}_+$. When these sets are finite, the result is a finite automaton with output. In any case, the functions involved are $\nu : Q \times I \to Q$ and $\eta : Q \times I \to O$. The "dynamics" of the automaton are given by

$$
\begin{aligned}
q_{k+1} &= \nu(q_k, i_k), \\
o_k &= \eta(q_k, i_k).
\end{aligned}
$$

Such a model is easily seen to encompass finite automata, Mealy and Moore machines, pushdown automata, Turing machines, Petri nets, etc.

§2.2 RELATED WORK

Here, we briefly mention other areas of inquiry related to hybrid systems. The interested reader is referred to the cited works for more details. Some connections, e.g., impulse control to hybrid systems control, are made clear later in the thesis.

§2.2.1 VARIABLE STRUCTURE AND SWITCHED SYSTEMS

Switched systems have been looked at by [30, 93, 119, 149] and others. We discuss them in §4.5 and §8. In **variable structure systems** [139], one takes a plant described by

$$
\dot{x} = f(x, u, t).
$$

Each component of the control is assumed to undergo discontinuity on an appropriate surface in the state space:

$$
u_i = \begin{cases} u_i^+(x, t), & \text{if } s_i(x) > 0, \\ u_i^-(x, t), & \text{if } s_i(x) < 0. \end{cases}
$$

The design problem is to choose the continuous functions u_i^+, u_i^-, and the functions $s_i(x)$ so that the system behaves in a desired manner (e.g., globally stabilization about some operating point).

§2.2.2 JUMP SYSTEMS

Also related to hybrid systems are so-called **jump systems**. The majority of results are for continuous-time linear systems with Markovian jumps. See [82, 101] and the references therein. These systems are modeled by

$$\dot{x}(t) = A(t, r(t))\, x(t) + B(t, r(t))\, u(t),$$

where $t \in [t_0, T]$, T may be finite or infinite, $x(t) \in \mathbb{R}^n (t)$ is the **x-process state**, $u(t) \in \mathbb{R}^m$ is the **x-process input**, and $A(t, r(t))$ and $B(t, r(t))$ are appropriately dimensioned real-valued matrices, which are functions of the random process $\{r(t)\}$. The **form process** $\{r(t)\}$ is a continuous-time discrete-state Markov process taking values in $\underline{N}$ with transition probability matrix $P \equiv \{p_{ij}\}$ given by

$$\begin{aligned} p_{ij} &= \Pr\{r(t + \Delta) = j \mid r(t) = i\}, \\ &= \begin{cases} \lambda_{ij}\Delta + O(\Delta), & \text{if } i \neq j, \\ 1 + \lambda_{ii}\Delta + O(\Delta), & \text{if } i = j, \end{cases} \end{aligned}$$

where $\Delta > 0$, $\lambda_{ij} \geq 0$ is the form transition rate from i to j $(i \neq j)$, and

$$\lambda_{ii} = -\sum_{j=1, j \neq i}^{N} \lambda_{ij}.$$

Note that, in general, A, B, and the λ_{ij}'s could be explicit functions of time. Also, the *switchings are nondeterministic* and *do not depend on the continuous state*.

In [60] a special class of hybrid systems is studied in which the Markov process above is replaced with a deterministic one:

$$\begin{aligned} \dot{x}(t) &= A(r(t))\, x(t) + B(r(t))\, u(t), \\ y(t) &= C(r(t))\, x(t), \end{aligned}$$

with $r(t) \in \underline{N}$. Stability, controllability, and observability results are given for the two cases described below. Let S_N denote the set of permutations of the values in $\underline{N}$ and numbers $\delta t_i > 0$ be given. The convention is that the ith system is active for time δt_i and the order of activation is given by S_N. Two cases are examined:

1. **Periodic.** The switching sequence is periodic of length N and is a repetition of some $s \in S_N$.

2. **General.** The switching sequence is a concatenation of members of S_N.

The stability results come directly from those for time-varying systems [150]. We note that in the case of controllability, for example, the rank of $N!$ matrices must be tested. Further, the restriction to constant δt_i is quite restrictive.

§2.2.3 SYSTEMS WITH IMPULSE EFFECT

Closely related to hybrid systems is the work on so-called **systems with impulse effect.** Most of these results appear only in the Russian literature, but fortunately there is an English monograph [10] which summarizes major results.

A **system of differential equations with impulse effect** is given by

1. the system of differential equations

$$\dot{x}(t) = f(t, x), \tag{2.3}$$

where $t \in \mathbb{R}$, $x \in \Omega \subset \mathbb{R}^n$, $f : \mathbb{R} \times \Omega \to \mathbb{R}^n$;

2. sets M_t, N_t of arbitrary topological structure contained in $\mathbb{R} \times \Omega$;

3. the operator $A_t : M_t \to N_t$.

The motion begins from (t_0, x_0) and moves along the curve $(t, x(t))$ described by Equation (2.3) with initial condition $x(t_0) = x_0$ until the instant $\tau_1 > t_0$ when it meets the set M_t. At time τ_1 the operator A_{τ_1} instantly transfers the point from position $(\tau_1, x(\tau_1))$ to $(\tau_1, x_1^+) \in N_{\tau_1}$, $x_1^+ = A_{\tau_1} x(\tau_1)$, from which the process continues to evolve under Equation (2.3) until it again encounters M_t, etc. Three classes of systems are considered in [10]:

- **Class I**: Systems with **fixed instants of impulse effect** are those for which M_t is represented by a sequence of hyperplanes $t = \tau_k$, where $\{\tau_k\}$ is a given sequence of instants of impulse effect. In this case, A_t is only defined for $t = \tau_k$, giving a sequence of operators $A_k : \Omega \to \Omega$.

- **Class II**: Systems with **mobile instants of impulse effect** are those for which M_t is represented by a sequence of hypersurfaces $\sigma_k \equiv t = \tau_k(x)$, $k \in \mathbb{Z}_+$. It is assumed that $\tau_k(x) < \tau_{k+1}(x)$ for $x \in \Omega$ and

$$\lim_{k \to \infty} \tau_k(x) = \infty$$

for all $x \in \Omega$. Again, A_t restricted to hypersurface σ_k is given by operator A_k.

- **Class III: Autonomous systems with impulse effect** are those where the sets $M_t \equiv M$ and $N_t \equiv N$ are subsets of Ω and $A_t \equiv A : M \to N$. Further, the equation $f(t, x)$ is replaced by $g(x)$.

Class III systems are closely related to hybrid systems, as we see below. Virtually all the results in [10], however, are for Class I and II systems.

§2.2.4 IMPULSE CONTROL

Impulse control, or **control by interventions**, deals with *choosing a discrete strategy for controlling a set of differential equations with minimal cost.* In particular, consider

$$\dot{x}(t) = f(x(t)), \qquad x(t_0) = x_0,$$

where $x(t) \in \mathbb{R}^n$, and let $\xi(t, x_0)$ denote its solution. There is a running cost $k : \mathbb{R}^n \to \mathbb{R}_+$ associated with the continuous state.

The discrete dynamics is modeled by a set-valued map $\Gamma : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ of **destinations**, each with an associated cost. Let $\mathrm{Gr}_\Gamma = \{(x,v) \mid x \in \mathbb{R}^n, v \in \Gamma(x)\}$. It is assumed that $\Gamma(x)$ is compact for each x and that the **jump cost** $c : \mathrm{Gr}_\Gamma \to \mathbb{R}_+$ is

1. bounded, with $c(x,\cdot)$ continuous on $\Gamma(x)$,

2. $c(x,v) + c(v,w) \geq c(x,w)$, for all $x \in \mathbb{R}^n$, $v \in \Gamma(x)$, and $w \in \Gamma(v)$.

3. $c(x,w) \geq c > 0$, for all $x \in \mathbb{R}^n$ and $w \in \Gamma(x)$.

A control policy π or **strategy** for this impulse control problem is a sequence $\{t_m; x_m, w_m\}$ of **intervention times**, **intervention states**, and **intervention destinations**, where $w_m \in \Gamma(x_m)$.

The trajectory under π is as follows: $x(t) = \xi(t, x_0)$ for $t < t_1$; at time t_1 the state is impulsively transferred from $x_1 \equiv \xi(t_1, x_0)$ to w_1 at a cost $c(x_1, w_1)$; from there, the process continues.

The infinite-time, discounted cost associated with the above problem is

$$J(\pi, x_0) = \int_0^\infty e^{-\alpha t} k(x(t)) \, dt + \sum_{t_m} e^{-\alpha t_m} c(x_m, w_m),$$

where $\alpha > 0$. The optimal control problem is to minimize $J(\pi, x_0)$ over all strategies. For more details see [51, 156]

§2.2.5 PIECEWISE DETERMINISTIC PROCESSES

A **piecewise-deterministic process** (PDP) taking values in an open set E of $\mathbb{R}^n$ with Borel σ-field $\mathcal{B}(E)$[4] and boundary ∂E is determined by [53]

1. the **flow** $\phi(t, x)$ in E satisfying $\phi(t, \phi(s, x)) = \phi(t + s, x)$;

2. the **jump rate** $\lambda : E \to \mathbb{R}_+$;

3. the **transition probability** $Q : \mathcal{B}(E) \times (E \cup \partial^* E) \to [0, 1]$ where

$$\partial^* E \equiv \{x \in \partial E \mid \phi(-t, z) \in E \text{ for } t \in (0, \epsilon) \text{ some } \epsilon > 0\}. \tag{2.4}$$

We define $t^*(x) \equiv \inf\{t > 0 \mid \phi(t, x) \in \partial^* E\}$ and $\Lambda(t, x) \equiv \int_0^t \lambda(\phi(s, x)) \, ds$ for $0 \leq t \leq t^*(x)$.

A sample, X_t starting from x, of the piecewise-deterministic process is constructed as follows. Pick a number T_1 from the distribution

$$P_x(T_1 > t) = \begin{cases} \exp(-\Lambda(t, x)), & t < t^*(x), \\ 0, & t \geq t^*(x). \end{cases}$$

Define $X_t = \phi(t, x)$ for $t < T_1$. Now independently choose an E-valued random variable Z_1 having distribution $Q(\cdot; \phi(T_1, x))$ and define $X_{T_1} = Z_1$. The process now restarts at Z_1. It is assumed that λ is bounded and the sequence of jump times $T_1, T_2, \ldots$ satisfies $\lim_{n \to \infty} T_n = \infty$ P_x-a.s.

---

[4]That is, $\mathcal{B}(E) = E \cap \mathcal{B}(\mathbb{R}^n)$.

§2.2.6 TIMED AND HYBRID AUTOMATA

There is a growing literature, mainly in computer science, that deals with hybrid systems as an outgrowth of automata theory. The main idea is to successively add time constraints on events and simple dynamics (such as clocks and timers) to finite automata in order to build on automata results. We do not attempt to summarize this literature and refer the interested reader to [1, 3, 63, 66, 98, 120].

This program comes full circle back to control theory in the work of Deshpande and Varaiya (also cf. [69]). A **hybrid automaton** [56, 57] is a system

$$H = (Q, \mathbb{R}^n, \Sigma, E, \Phi, \Gamma),$$

where Q is the finite set of **discrete states**, $\mathbb{R}^n$ is the the set of **continuous states**, and Σ is the finite set of **discrete events**. The finite set of **edges**,

$$E \subset Q \times 2^{\mathbb{R}^n} \times \Sigma \times \{\mathbb{R}^n \to \mathbb{R}^n\} \times Q.$$

models the discrete event dynamics of the system. An edge

$$E \ni e = (q_e, X_e, V_e, r_e, q_e')$$

is enabled when the discrete state is in q_e and the continuous state is in X_e. When the transition is taken, the event $V_e \in \Sigma$ is accepted, the continuous state is reset according to map r_e, and the system enters discrete state q_e'. The reset map is allowed to be set valued (denoted R_e), in which case the continuous state is reset nondeterministically when e is taken. The **continuous dynamics** is given by the set of controlled vector fields

$$\Phi = \{F_q : \mathbb{R}^n \to 2^{\mathbb{R}^n} \setminus \{\emptyset\} \mid q \in Q\}.$$

When the discrete state is q, the continuous state evolves according to the **differential inclusion**

$$\dot{x}_c(t) \in F_q(x_c(t)).$$

Finally,

$$\Gamma = \{\Gamma_q \subset \mathbb{R}^n \mid q \in Q\}$$

is a set of invariance conditions that the system state must satisfy, that is, when the system is in phase q, the continuous state is required to be in Γ_q.

Part I

Modeling of Hybrid Systems

Chapter 3

Hybrid Phenomena and Models

In this chapter, we identify the discrete phenomena which occur in hybrid systems. We then review in detail several models of hybrid systems from the control and dynamical systems literature and draw relations among them.

§3.1 INTRODUCTION

A hybrid system is a system that involves continuous states and dynamics, as well as some **discrete phenomena** corresponding to discrete states and dynamics. In this thesis, our focus is on the case where the continuous dynamics is given by a differential equation

$$\dot{x}(t) = \xi(t), \qquad t \geq 0. \tag{3.1}$$

Here, $x(t)$ is the **continuous component** of the state taking values in some subset of a Euclidean space. $\xi(t)$ is a vector field which generally depends on $x(t)$ and the aforementioned discrete phenomena.

Hybrid control systems are control systems that involve continuous states, dynamics, and controls, as well as discrete phenomena corresponding to discrete states, dynamics, and controls. Here, $\xi(t)$ is a *controlled* vector field which generally depends on $x(t)$, the *continuous component* $u(t)$ of the control policy, and the aforementioned discrete phenomena.

The chapter is organized is follows. First, we begin by identifying the discrete phenomena that generally arise in hybrid systems. We identify four types:

1. autonomous switching,

2. autonomous jumps (also called autonomous impulses),

3. controlled switching,

4. controlled jumps (also called controlled impulses).

In the next section, we briefly examine these discrete phenomenon and give examples of each. We also discuss how digital automata may be viewed as evolving in continuous time, which sets the stage for their interacting with ODEs below.

Next, we review in turn several models of hybrid systems developed from the control and dynamical systems point of view. For sure, there are many others [3, 4, 66, 120]. The models here have been chosen as much for the clarity and rigor of their presentation as for the mechanisms they use to combine discrete and continuous dynamics. Specifically, we review the following models of hybrid systems, in order of (original) appearance of the cited papers:

1. Witsenhausen's model [152, §3.3],

2. Tavernini's model [135, §3.3],

3. Back-Guckenheimer-Myers model [9, §3.5],

4. Nerode-Kohn model [114, §3.6],

5. Antsaklis-Stiver-Lemmon model [5, §3.7],

6. Brockett's models [38, §3.8].

For further discussion and examples, the reader is referred to the original papers.

Some models in the papers above allow time-varying vector fields, but we only consider time-invariant ones here. Also, we have sometimes changed notation from the original papers to make the presentation more uniform and place the models in as similar a light as possible. In §§3.9–3.10 we briefly discuss and compare the six models.

§3.2 HYBRID PHENOMENA

§3.2.1 AUTONOMOUS SWITCHING

Autonomous switching is the phenomenon where the *vector field $\xi(\cdot)$ changes discontinuously when the continuous state $x(\cdot)$ hits certain "boundaries"* [5, 114, 135, 152]. The simplest example of this is when it changes depending on a "clock" which may be modeled as a supplementary state variable [38]. An example of autonomous switching is the following.

Example 3.1 (Hysteresis) *Consider the following model of a system with hysteresis* [135]:

$$
\begin{aligned}
\dot{x}_1 &= x_2 - \phi(x_1), \\
\dot{x}_2 &= H(\psi(x_1, x_2)) - \phi(x_2),
\end{aligned}
$$

where the multi-valued function H is shown in Figure 3-1. The functions ϕ, ψ depend on the exact system under consideration.
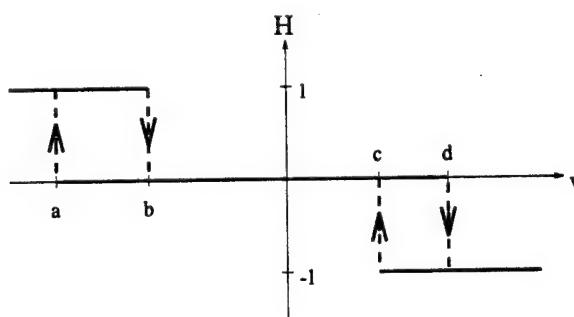
Figure 3-1: Hysteresis function.

Note that this system is not just a differential equation whose right-hand side is piecewise continuous. There is "memory" in the system, which affects the value of the vector field. Indeed, such a system naturally has a finite automaton associated with the function H, as pictured in Figure 3-2.
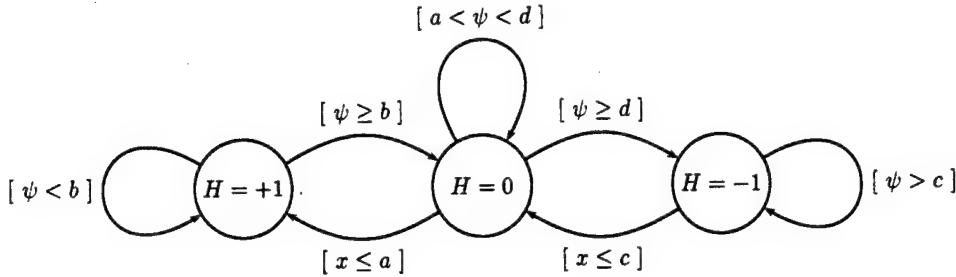
Figure 3-2: Finite Automaton Associated with Hysteresis Function.

§3.2.2 AUTONOMOUS JUMPS

An **autonomous jump** is the phenomenon where the *continuous state* $x(\cdot)$ *jumps discontinuously on hitting prescribed regions of the state space* [9, 10]. We may also call these **autonomous impulses**. The simplest examples possessing this phenomenon are those involving collisions.

Example 3.2 (Collisions) *Consider the case of the vertical and horizontal motion of a ball of mass m in a room under gravity with constant g (see Figure 3-3). In this case, the dynamics are given by*

$$
\begin{aligned}
\dot{x} &= v_x, \\
\dot{y} &= v_y, \\
\dot{v}_x &= 0, \\
\dot{v}_y &= -mg.
\end{aligned}
$$

Further, upon hitting the boundaries $\{(x,y) \mid y = 0 \text{ or } y = C\}$ we instantly set v_y to $-\rho v_y$, where $\rho \in [0,1]$ is the coefficient of restitution. Likewise, upon hitting $\{(x,y) \mid x = 0 \text{ or } x = R\}$ v_x is set to $-\rho v_x$.

§3.2.3 CONTROLLED SWITCHING

Controlled switching is the phenomenon where the *vector field $\xi(\cdot)$ changes abruptly in response to a control command*, usually with an associated cost. This can be interpreted as switching between different vector fields [156]. Controlled switching arises, for instance, when one is allowed to pick among a number of vector fields:

$$
\dot{x} = f_i(x), \qquad i \in Q \simeq \{1, 2, \ldots, N\}.
$$

Example 3.3 (Satellite Control) *As a simple example of satellite control consider*

$$
\ddot{\theta} = \tau_{\text{eff}} v,
$$

where θ is angular position, $\dot{\theta}$ angular velocity of the satellite, and $v \in \{-1, 0, 1\}$ depending on whether the reaction jets are full reverse, off, or full on.
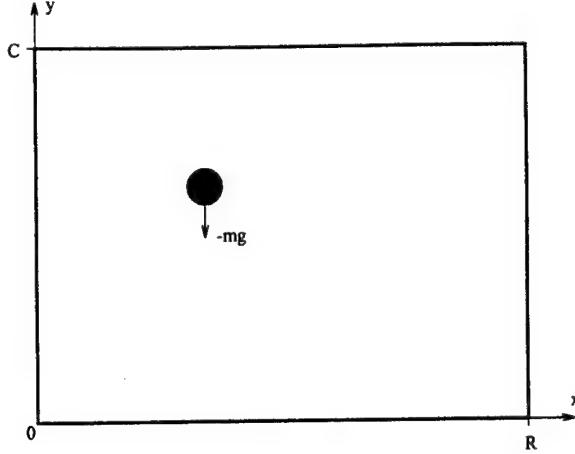
Figure 3-3: Ball bouncing in an enclosed room.

An example that includes controlled switching and continuous controls is the following.

Example 3.4 (Transmission) *Consider a simplified model of a manual transmission, modified from one in* [38]:

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = [-a(x_2/v) + u]/(1+v),$$

where x_1 is the ground speed, x_2 is the engine RPM, $u \in [0,1]$ is the throttle position, and $v \in \{1,2,3,4\}$ is the gear shift position. The function a is positive for positive argument.

§3.2.4 CONTROLLED JUMPS

An **controlled jump** is the phenomenon where the *continuous state $x(\cdot)$ changes discontinuously in response to a control command*, usually with an associated cost [14]. We may also call these **controlled impulses**. An example is the following.

Example 3.5 (Inventory Management) *In a simple inventory management model* [14], *there are a "discrete" set of restocking times $\theta_1 < \theta_2 < \cdots$ and associated order amounts $\alpha_1, \alpha_2, \ldots$. The equations governing the stock at any given moment are*

$$\dot{y}(t) = -\mu(t) + \sum_i \delta(t - \theta_1)\alpha_i$$

where μ represents degradation or utilization dynamics and δ is the Dirac delta function.

> **NOTE.** If one makes the stocking times and amounts an explicit function of y (or t), then these controlled jumps become autonomous jumps.

§3.2.5 DIGITAL AUTOMATA AND ODEs

Usually, digital automata [§2.1.4] are thought of as evolving in "abstract time," where only the ordering of symbols or "events" matters:

$$q_{k+1} = \nu(q_k, i_k),$$
$$o_k = \eta(q_k, i_k).$$

We may add the notion of time by associating with the kth transition the time at which it occurs:

$$q(t_{k+1}) = \nu(q(t_k), i(t_k)),$$
$$o(t_k) = \eta(q(t_k), i(t_k)).$$

Finally, this automaton may be thought of as operating in "continuous time" by the convention that the state, input, and output symbols are piecewise right- or left- continuous functions.

NOTE. The notation t^- may used to indicate that the finite state is piecewise continuous from the right:

$$q(t) = \nu(x(t), q(t^-))$$

Likewise

$$q(t^+) = \nu(x(t), q(t))$$

denotes that it is piecewise-continuous from the left. To avoid making the distinction here we use Sontag's more evocative discrete-time transition notation [133]

$$q^+(t) = \nu(x(t), q(t))$$

to denote the "successor" of $q(t)$. Its "predecessor" is denoted $q^-(t)$. This notation makes sense since no matter which convention is used for $q(t)$'s piecewise continuity, we still have $q^+(t) = q(t^+)$.

The result is the following system of equations, where q, o are piecewise continuous in time:

$$q^+(t) = \nu(q(t), i(t)),$$
$$o(t) = \eta(q(t), i(t)). \tag{3.2}$$

Here, the state $q(t)$ changes only when the input symbol $i(t)$ changes. Also, note that this reduces to the previous automaton equation

$$q^+[t] = \nu(q([t]), i[t]),$$
$$o[t] = \eta(q[t], i[t]),$$

where $[t]$ denotes the time at which the input symbol last changed. Thus, we have the idea of an automaton whose update times are not the abstract members of $\mathbb{Z}_+$, but the event times in $\mathbb{R}$ when the input symbol i changes. Note that in the usual case, a finite automaton can be presented with the same input symbol for two successive time intervals. These situations must be handled differently (e.g., by adding new states and symbols) in the continuous-time version.

Finally, we may wish to model the interaction of digital automata with ODEs. To

accomplish this we note that such equations can be thought of as a special case of ODEs with controlled jumps (§4):

$$
\begin{aligned}
\dot{q}(t) &= 0, \\
q^+(t) &= \nu(q(t), i(t)), \\
o(t) &= \eta(q(t), i(t)).
\end{aligned}
$$

Suppose these equations interact with a set of differential equations with output $i(t)$. Thus, the symbol $i(t)$ jumps as a function of the state $x(t)$ of the differential equation

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t)), \\
i(t) &= h(x(t)).
\end{aligned}
$$

At these interaction times, the discrete phenomena are autonomous jumps in the combined system. We may also allow u to depend on $o(t)$, which is the case in hybrid systems below.

§3.3 WITSENHAUSEN'S MODEL

In [152], Witsenhausen introduces a class of continuous-time systems with part continuous, part discrete state—in short, what we would call a continuous-time hybrid system. The class of systems he considers is restricted by the following conditions:

1. At a transition, that is, at a time when the discrete state undergoes a change, the state vector is still continuous, though the vector field may change discontinuously. No jumps of the continuous state vector are allowed.

2. Transitions occur when and only when the continuous state vector satisfies a condition given for each type of transition. In the case of control inputs, these influence transitions only through the differential equations, never directly.

3. Some technical requirements on the data describing the system.

Witsenhausen starts with continuous systems, described by

$$
\dot{x}(t) = f(x(t), u(t)),
$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$. The effect of the discrete components of the state on these continuous components is accounted for by letting f depend on the discrete state:

$$
\dot{x}(t) = f(q(t), x(t), u(t)).
$$

The coupling in the opposite direction is described by conditions on the continuous state for which transitions of the discrete component occur. In particular, let $Q \subset \mathbb{Z}$ be finite. The state of the system is characterized at any time by the pair (q, x) ranging over $Q \times \mathbb{R}^n$; q is called the **discrete state**, and x is the **continuous state**.

A transition of the discrete state from $q = i$ to $q = j \neq i$ is **triggered** when the continuous state x reaches a given **transition set** $M_{i,j} \subset \mathbb{R}^n$. There is one such set, possibly empty, for each ordered pair of distinct indexes from Q.

Define the **arrival set**

$$
M_i^+ \equiv \bigcup_{j \neq i} M_{j,i}.
$$

That is, the set of all values of the continuous state for which a transition into discrete state i can occur from some other discrete state (in our parlance, the set of destinations of autonomous switchings into discrete state i). Likewise, define the **departure set**

$$M_i^- \equiv \bigcup_{j \neq i} M_{i,j}.$$

That is, the set of all values of the continuous state for which a transition from discrete state i into some other discrete state can occur (in our parlance, the set of autonomous switchings from discrete state i).

Witsenhausen places three key assumptions on the sets above, which we explore in some detail:

1. For any three distinct indexes i, j, k from Q, the sets $M_{i,j}$ and $M_{i,k}$ are disjoint.

2. For all $i \in Q$, M_i^- is closed in $\mathbb{R}^n$ (equipped with the usual topology).

3. For all $i \in Q$, the sets M_i^+ and M_i^- are disjoint in $\mathbb{R}^n$.

The first assumption says that it is never the case that the conditions for transition to two or more different discrete states are fulfilled. The second assumption is more technical; its usefulness is seen in the description of the dynamics below. The third assumption guarantees that after a discrete transition is made, the condition for a further, instantaneous transition to some other discrete state does not occur. This rules out "transition loops," in which an infinite number of discrete transitions occur in a single instant. If there is no loop, and a transition sequence starting at (x, i_0) goes through $(x, i_1), \ldots$ and terminates at some state (x, i^*), then the whole sequence can be avoided by considering $x \in M_{i_0, i_*}$ instead of M_{i_0, i_1}.

Witsenhausen assumes that each $f(q, x, u)$ is continuous in (x, u) for fixed q and continuously differentiable in x for fixed (q, u). He restricts the set of controls to be as follows. Let t_f be a variable end time in $[t_0, \infty)$ and $\Omega \subset \mathbb{R}^m$, closed. A control is a pair $[t_f, u(\cdot)]$ where $u(\cdot)$ is any piecewise continuous function from $[t_0, t_f]$ into Ω.

We are now ready to describe the evolution of the system from initial state (x_0, q_0) at time t_0 until (x_f, q_f) at time t_f, under the control $[t_f, u(\cdot)]$:

1. Solve

$$\dot{x}(t) = f(q_0, x(t), u(t)), \qquad x(t_0) = x_0.$$

The conditions on f assure the existence of a unique solution either up to t_f or up to some earlier escape time. In either case a continuous path, $x_0(\cdot)$, in $\mathbb{R}^n$ is given. If this path does not meet $M_{q_0}^-$, then it gives the evolution of the continuous state and the discrete state remains constant at q_0.

2. If the path $x_0(\cdot)$ meets $M_{q_0}^-$ at some point x^* at time t^*, then by virtue of $x_0([t_0, t^*])$ compact and $M_{q_0}^-$ closed (Assumption 2), there exists an earliest time of intersection t_1 with corresponding point x_1. By Assumption 1, x_1 belongs to exactly one of the transition sets in $M_{q_0}^-$, say, M_{q_0, q_1}. The state of the system at time t_1 is defined to be (x_1, q_1). Note that $t_1 > t_0$, since otherwise (x_0, q_1) would have been considered the initial state.

3. The point x_1 now belongs to $M_{q_1}^+$ and (by Assumption 3) not to $M_{q_1}^-$. Thus, we may repeat the above steps from (x_1, q_1). Furthermore, if a second transition time t_2 exists, it satisfies $t_2 > t_1$.

Thus, the above rules, applied recursively, generate a unique state evolution $[x(\cdot), q(\cdot)]$ up to t_f, some earlier escape time, or a **critical time** representing a point of accumulation of transition times. See Figure 3-4.

Finally, Witsenhausen notes that since the transitions of $x(\cdot)$ always take place on the boundary ∂M_i^- of a departure set (which is closed by Assumption 2), all transition sets $M_{i,j}$ can be replaced by their **reductions** $M_{i,j} \cap \partial M_i^-$. He also gives some optimal control results, which are discussed in §10. We refer to the above as the WHS model, for Witsenhausen hybrid systems.

§3.4 TAVERNINI'S MODEL

Tavernini introduces and discusses so-called *differential automata* in [135]. He was motivated to study such systems as a means of modeling hysteretic phenomena like backlash and friction (cf. Example 3.1).

A **differential automaton**, A, is a system (S, f, ν) where

- S is the state-space of A, $S = \mathbb{R}^n \times Q$, $Q \simeq \{1, 2, \ldots, N\}$ is the **discrete state space** of A, and $\mathbb{R}^n$ is the **continuous state space** of A;

- f is a finite family $f(\cdot, q) : \mathbb{R}^n \to \mathbb{R}^n$, $q \in Q$, of vector fields, the **continuous dynamics** of A;

- $\nu : S \to Q$ is the **discrete transition function** of A.

Let $\nu_q \equiv \nu(\cdot, q)$, $q \in Q$. Define $I(q) = \nu_q(\mathbb{R}^n) \backslash \{q\}$, that is, the set of discrete states "reachable in one step" from q. We require that for each $q \in Q$ and each $p \in I(q)$ there exist closed sets

$$M_{q,p} \equiv \nu_q^{-1}(p).$$

The sets $\partial M_{q,p}$ are called the **switching boundaries** of the automaton A. Define $M_q = \bigcup_{p \in I(q)} M_{q,p}$ and define the **domain of capture** of state q by

$$C(q) \equiv \mathbb{R}^n \backslash M_q = \{x \in \mathbb{R}^n \mid \nu(x, q) = q\}.$$

The equations of motion are

$$\begin{aligned} \dot{x}(t) &= f(x(t), q(t)), \\ q^+(t) &= \nu(x(t), q(t)), \end{aligned}$$

with initial condition $[x(0), q(0)]^T \in \bigcup_{q \in Q} C(q) \times \{q\}$. The notation t^- indicates that the discrete state is piecewise continuous from the right. Thus, starting at $[x_0, i]$, the continuous state trajectory $x(\cdot)$ evolves according to $\dot{x} = f(x, i)$. If $x(\cdot)$ hits some $\partial M_{i,j}$ at time t_1, then the state becomes $[x(t_1), j]$, from which the process continues. See Figure 3-4.

Tavernini places restrictions on the model above: First each $f(\cdot, q)$, $q \in Q$, is assumed to be globally Lipschitz so that the continuous dynamics are well-behaved. Also, for each $q \in Q$ and $p \in I(q)$, the set $M_{q,p}$ is required to be connected and there must exist a function $g_{q,p} \in C^1(\mathbb{R}^n, \mathbb{R})$ with 0 in its image a regular value such that

$$M_{q,p} = \{x \in \mathbb{R}^n \mid g_{q,p}(x) \geq 0\}.$$
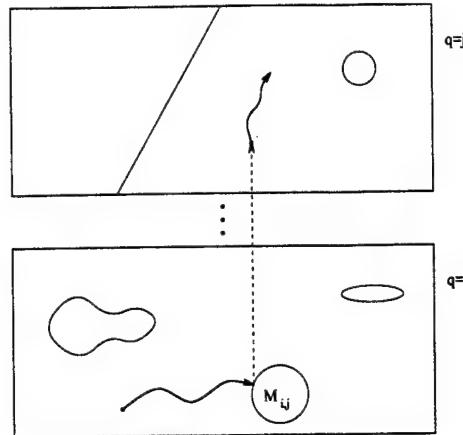
Figure 3-4: Example dynamics of Tavernini's model.

Thus, $\nu_q^{-1}(p)$ is an n-submanifold of $\mathbb{R}^n$ with boundary

$$\partial M_{q,p} = \{x \in \mathbb{R}^n \mid g_{q,p}(x) = 0\},$$

which is an $(n-1)$-submanifold of $\mathbb{R}^n$.

Finally, [135] makes the following three key assumptions on differential automata:

- Define $\alpha_q = \min\{\text{dist}(M_{q,p}, M_{q,p'}) \mid p, p' \in I(q), p \neq p'\}$. We require that

$$\alpha(A) \equiv \min_{q \in Q} \alpha_q > 0$$

be satisfied. That is, the distance between any two sets with different discrete transitions is bounded away from zero.

- Define $\beta_{q,p} = \min\{\text{dist}(\partial M_{q,p}, \partial M_{p,p'}) \mid p' \in I(p)\}$. We require that the inequality

$$\beta(A) \equiv \min_{q \in Q} \min_{p \in I(q)} \beta_{q,p} > 0$$

be satisfied. That is, after a discrete transition, the next set from which another discrete transition takes place is at least a fixed distance away.

- The assumption on $\alpha(A)$ is such that $C(q)$ is an open set with boundary $\partial C(q) = \partial M_q = \bigcup_{p \in I(q)} \partial M_{q,p}$. We require that the inclusions

$$\partial M_{q,p} \subset C(p), \qquad p \in I(q), q \in Q$$

be satisfied. That is, after a discrete transition one is found in an open set on which the dynamics are well-defined.

With these assumptions,[1] Tavernini proves that the initial value problem has a unique solution with finitely many switching points. Let $[x(\cdot), q(\cdot)]$ denote the solution corresponding to the initial value $[x_0, q_0]$. Then q defines a sequence of discrete states $q_0, q_1, q_2, \ldots,$

---

[1] Actually, the vector fields $f(\cdot, q)$ and switching functions $g_{q,p}$ are assumed to be smooth.

with switching points $t_1, t_2, \ldots$, where t_i denotes the time of transition from q_{i-1} to q_i. If $[x'(\cdot), q'(\cdot)]$ denotes the solution when the initial value is $[x'_0, q_0]$ where x'_0 is near x_0, then we should have $[x'(\cdot), q'(\cdot)]$ "near" $[x(\cdot), q(\cdot)]$ in the sense that q' should define the same "discrete trajectory" $q_0, q_1, q_2, \ldots$ with possibly different switching points. However, the corresponding switching points of the two solutions should be close, i.e., $|t'_i - t_i|$ should be "small" whenever $|x'_0 - x_0|$ is "small." Tavernini defines a topology to make this precise. He also shows that the set of points with such a property is an open, dense subset of $C(q_0)$, denoted S^0.

Finally, Tavernini concentrates on the analysis of numerical approximations of the trajectories of differential automata. Briefly, if $(x_0, q_0) \in S^0$ then the result of numerical integration of the trajectory starting from x'_0 uniformly approaches that of the differential automaton starting from x_0 as the integration step size plus $d(x_0, x'_0)$ goes to zero. See [135] for details.

We refer to the above as the TDA model, for Tavernini's differential automata.

§3.5 BACK-GUCKENHEIMER-MYERS MODEL

The framework proposed by Back, Guckenheimer, and Myers in [9] is similar in spirit to the Tavernini model. The model is more general, however, in allowing "jumps" in the continuous state-space and setting of parameters when a switching boundary is hit. This is done through **transition functions** defined on the switching boundaries. Also, the model allows a more general state space.

More specifically, the model consists of a state space

$$X = \bigcup_{q \in Q} X_q, \qquad Q \simeq \{1, \ldots, N\},$$

where each X_q is a connected, open set of $\mathbb{R}^n$. Notice that the sets X_q are not required to be disjoint.

The continuous dynamics are given by vector fields $f_q : X_q \to \mathbb{R}^n$. Also, one has open sets U_q such that $\overline{U}_q \subset X_q$ and ∂U_q is piecewise smooth. For $q \in Q$, the transition functions

$$G_q : X_q \to X \times Q$$

govern the jumps that take place when the state in X_q hits ∂U_q. They must satisfy $\pi_1(G_q(x)) \in \overline{U}_{\pi_2(G_q(x))}$, where π_k is the kth coordinate projection function. Thus, $\pi_1(G_q(x))$ is the "continuous part" and $\pi_2(G_q(x))$ is the "discrete part" of the transition function.

The dynamics are as follows. The state starts at point x_0 in U_i. It evolves according to $\dot{x} = f_i(x)$. If $x(\cdot)$ hits some ∂U_i at time t_1, then the state instantaneously jumps to state ξ in $\overline{U}_j$, where $G_i(x(t_1)) = (\xi, j)$. From there, the process continues. We refer to this as the BGM model. See Figure 3-5, which is taken from [9].

As in [135], it is assumed in [9] that the switching boundaries are fairly regular. In particular, it is assumed that the **switching boundaries** ∂U_q have a concrete representation in terms of the zeros of

$$h_q \equiv \min\{h_{q,1}, \ldots, h_{q,N_q}\}.$$

where the $h_{q,i} : X_q \to \mathbb{R}$ are smooth. The convention then is such that $h_q > 0$ on U_q. Thus,
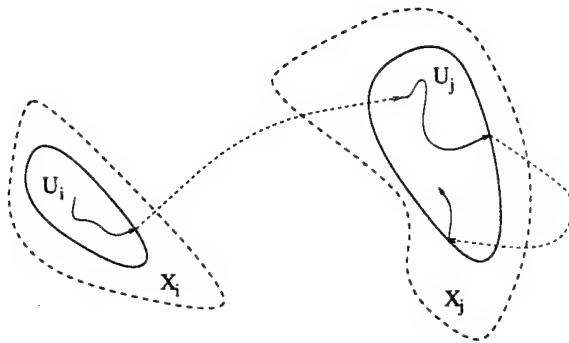
Figure 3-5: Example dynamics of Back-Guckenheimer-Meyers model. Reproduced from [9].

the switching boundaries are $(n-1)$-dimensional Lipschitz continuous manifolds.

> NOTE. This does not add much power (over single functions) since Lipschitz functions are strongly approximated by C^1 functions: for every $\epsilon > 0$ a C^1 function can be chosen that coincides with a Lipschitz function except on a set of measure ϵ [109].

The model above is fairly expressive, allowing the modeling of a large variety of phenomena. However, its expressiveness does allow the possibility of some seemingly "anomalous" behavior. For example, since one allows jumping to the boundary of the sets U_i, trajectories may infinitely "cycle" if $G_j(x) = (y, i)$ and $G_i(y) = (x, j)$. In a simulation facility, however, such conditions can presumably be detected and reported to the user for interpretation.

The paper [9] presents computer tools that have been developed by its authors for the simulation of hybrid systems. As an example, Raibert's one-legged hopping robot [122] is placed into their framework.

We refer to the above model (simplified from the one in [9]) as the BGM model.

§3.6 NERODE-KOHN MODEL

In [114], Nerode and Kohn take an automata-theoretic approach to systems composed of interacting ODEs and FA. The basic philosophy of the models discussed in [114] is given in great generality, with a subsequent specialization to various cases, e.g., deterministic versus non-deterministic. To keep the discussion germane to that so far, we discuss here the so-called "event-driven, autonomous sequential deterministic model" [114, p. 331]. We refer to it as the NKSD (for sequential deterministic) model. Here, autonomous refers to the fact that the ODEs do not explicitly depend on time, although this is without loss of generality by appending to the state a single equation for t.

The model consists of three basic parts: plant, digital control automaton, and interface. In turn, the interface is comprised of an analog-to-digital (AD) converter and digital-to-analog (DA) converter. See Figure 3-6.

The **plant** is modeled as in Equation (2.2).[2] It is considered to be an input/output automaton in the following sense. The states of the system (in this sequential deterministic case) are merely the usual plant states, members of $\mathbb{R}^n$ [114, p. 333]. The input alphabet is

---

[2]We have lumped the control and disturbance signals of [114] into a single signal u.
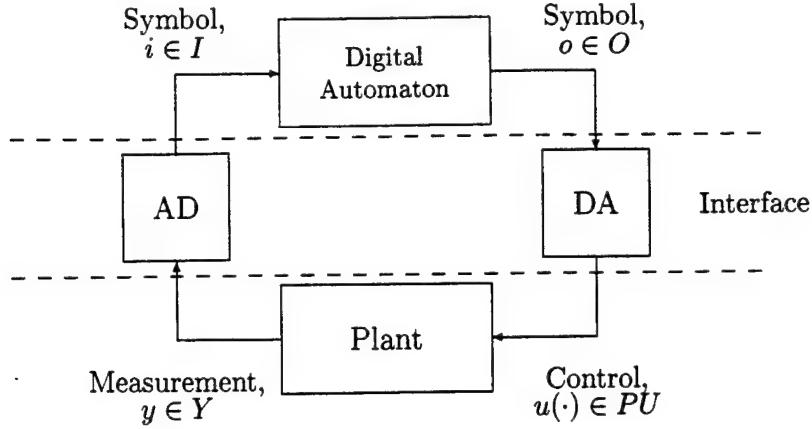
Figure 3-6: Hybrid system as in Nerode-Kohn model.

formally taken to be the set of members of $(u(\cdot), \delta_k)$ where δ_k is a positive scalar and $u(\cdot)$ is a member of the set of piecewise right-continuous functions in $U^{[0,\infty)}$. Let PU, for piecewise U, denote the latter set. Suppose the plant is in state x_k at time t_k. The "next state" of the transition function from this state with input symbol $(u(\cdot), \delta_k)$ is given by $x_{k+1} \equiv x(t_k + \delta_k)$, where $x(\cdot)$ is the solution on $[t_k, t_k + \delta_k]$ of

$$\dot{x}(t) = f(x(t), u(t - t_k)), \qquad x(t_k) = x_k.$$

Setting $t_{k+1} = t_k + \delta_k$, the process is continued.

The **digital control automaton** is a digital automaton as discussed in §2.1.4 and §3.2.5. In general, then, Q, I, and O are each isomorphic to subsets of $\mathbb{Z}_+$. However, the interesting case is where these sets are finite, which is discussed below. As noted before, this automaton may be thought of as operating in "continuous time" by the convention that the state, input, and output symbols are piecewise right-continuous functions, leading to Equation (3.2).

It remains to couple these two "automata." This is done through the **interface** consisting of the following two maps:

- The **analog-to-digital map** $AD : Y \times Q \to I$,

- The **digital-to-analog map** $DA : O \to PU$.

The AD symbols are determined by (FA-state-dependent) partitions of the output space Y. These partitions are not allowed to be arbitrary, but are the "essential parts" of small topologies placed on Y for each $q \in Q$. We explain this later. To each $o \in O$ is associated an open set of PU. The DA signal corresponding to output symbol o is chosen from this open set of plant inputs. The scalar δ_k is a formal construct, denoting the time until the next "event." It is not actually computed or chosen by the digital automaton, nor is it actively used by the plant in computing its update equations.

The dynamics of the above model are then similar to those of the Tavernini model. Two important distinctions arise: input and output for both the ODEs and FA have been included, and the maps AD and DA have been added. Specifically, we have

$$\dot{x}(t) \quad = \quad f[x(t), DA(o(t), t - [t])],$$

$$
\begin{aligned}
y(t) &= h[x(t)], \\
q^+(t) &= \nu[q(t), AD(y(t), q(t))], \\
o(t) &= \eta[q(t), AD(y(t), q(t))].
\end{aligned}
$$

Briefly, the combined dynamics is as follows. Assume the continuous state is evolving according to the first equation and that the FA is in state q. Then $AD(\cdot, q)$ assigns to output $y(t)$ a symbol from the input alphabet of the FA. When this symbol changes, the FA makes the associated state transition, causing a corresponding change in its output symbol o. Associated with this symbol is a control input, $DA(o)$, which is applied as input to the differential equation until the input symbol of the FA again changes.

Now, we explain what is meant by the "small topologies" mentioned above, concentrating on the AD map. Nerode and Kohn introduce topologies that make each mapping $AD_q \equiv AD(\cdot, q)$, $q \in Q$, continuous (see Algorithm 6.2). The sets $AD_q^{-1}(i)$, $i \in I$ are the essential parts mentioned above. For a verification that AD_q is continuous, as well as other results on AD and DA maps, see [24, §6].

The starting point of the Nerode-Kohn approach is an assumption that one can only realistically distinguish points up to knowing the open sets in which they are contained. That is what led them to use the small topologies above to encode the plant output symbols. However, the bottom line is that by combining information of inclusion in different open sets, the AD_q functions, $q \in Q$, form partitions of the measurement space. Although the small topologies are meant to provide "reasonable partitions," it is interesting to note that one can still "identify" single points in the model.

> **EXAMPLE.** Consider as a representative example zero in $[-1, 1]$. Then the open sets $[-1, 1]$, $[-1, 0)$, and $(0, 1]$ give information to exactly deduce $x = 0$. Such anomalies lead to a breakdown of the description of the dynamics above in the sense that it is easy to construct examples where the formal input letter to the plant is $(u, 0)$.

The Nerode-Kohn paper develops the underpinning of a theoretical framework for the hybrid continuous/rule-based controllers used by Kohn in applications. Continuity in the small topologies associated with the AD and DA maps above plays a vital role in the theory of those controllers. See [114] and the references therein for details.

§3.7 ANTSAKLIS-STIVER-LEMMON MODEL

In [5], Antsaklis, Stiver, and Lemmon take a discrete-event dynamical systems (DEDS, [74]) approach to hybrid systems. Conceptually, the model is related to that of Nerode-Kohn, but we quickly review it here. We refer to it as the ASL model.

Like the NKSD model, the ASL model consists of three basic parts: the plant, the controller, and the interface. See Figure 6-1. The **plant** is modeled as in Equation (2.2). The **controller** is a discrete event system, modeled as a digital automaton. We think of it as operating in continuous time as in §3.2.5:

$$
\begin{aligned}
q^+(t) &= \nu(q(t), i(t)), \\
o(t) &= \eta(q(t)),
\end{aligned}
$$

where $q(t) \in Q$, $i(t) \in I$, and $o(t) \in O$, the state space, plant symbols, and controller symbols, respectively. The sets Q, I, and O are unspecified in [5], but we take from context

that they are each isomorphic to subsets of $\mathbb{Z}_+$. The maps are $\nu : Q \times I \to Q$ and $\eta : Q \to O$. The subscript k denotes the kth symbol in a sequence. The output map does not depend on the current symbol, which is without loss of generality after adding more states.

The plant and controller communicate through an **interface** consisting of two memoryless maps, AD and DA. The first map, called the **actuating function**, $DA : O \to \mathbb{R}^m$, converts a controller symbol to a piecewise constant plant input:

$$u(t) = DA(o(t)).$$

The second map, called the plant symbol **generating function**, $AD : \mathbb{R}^n \to I$, is a function which maps the plant state space to the set of plant symbols as follows

$$i(t) = AD(x(t)).$$

The function AD is based upon a partition of the state space, where each element of the partition is associated with one plant symbol. The combined dynamics is similar to that of the NKSD model.

The model is simple but fairly general. The fact that arbitrary partitions are allowed limits what one can prove about the trajectories of this model. Several example systems are given in [5]. Results, mainly from the DEDS point of view, may be found in [5] and the references therein.

§3.8 BROCKETT'S MODELS

Several models of hybrid systems are described in [38]. We only discuss those which combine ODEs and discrete phenomena since that is our focus here. Two models combining difference equations and discrete phenomena are also discussed in [38].

The first model, which Brockett calls a **type B hybrid system**, is as follows:

$$\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), v\lfloor p\rfloor), \\
\dot{p}(t) &= r(x(t), u(t), v\lfloor p\rfloor),
\end{aligned}$$

where $x(t) \in X \subset \mathbb{R}^n$, $u(t) \in U \subset \mathbb{R}^m$, $p(t) \in \mathbb{R}$, $v\lfloor p\rfloor \in V$, $f : \mathbb{R}^n \times \mathbb{R}^m \times V \to \mathbb{R}^n$, and $r : \mathbb{R}^n \times \mathbb{R}^m \times V \to \mathbb{R}$. Here, X and U are open subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, and V is isomorphic to a subset of $\mathbb{Z}_+$. Also, the **rate equation** r is required to be nonnegative for all arguments, but need have no upper bound imposed upon it. We denote such a system as BB, short for Brockett's type B model.

Brockett has mixed continuous and "symbolic" controls by the inclusion of the special **counter** variable p. The control $u(t)$ is the continuous control exercised at time t; the control $v\lfloor p\rfloor$ is the pth symbolic or discrete control, which is exercised at the times when p passes through integer values. In general, one may also introduce (as in [38]) continuous and symbolic output maps:

$$\begin{aligned}
y(t) &= c(x(t), v\lfloor p\rfloor), \\
o\lfloor p\rfloor &= \eta(y[t], v\lfloor p\rfloor).
\end{aligned}$$

In this case, one may limit f by allowing it to depend only on y instead of the full state x. Note, we have used $[t]$ to denote the value of t at which p most recently became an integer.

Brockett also introduces a **type D hybrid system** as follows:

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), z\lfloor p \rfloor), \\
\dot{p}(t) &= r(x(t), u(t), z\lfloor p \rfloor), \\
z\lceil p \rceil &= \nu(x[t], z\lfloor p \rfloor, v\lfloor p \rfloor),
\end{aligned}
$$

where $z \in Z$, and Z is isomorphic to a subset of $\mathbb{Z}_+$. Here, $\nu : \mathbb{R}^n \times Z \times V \to Z$, with all other definitions as above except that Z replaces V in those for f and r. Again, u and v are the continuous and discrete controls, respectively. We denote such a system by BD. We may picture the dynamics as in Figure 3-7.
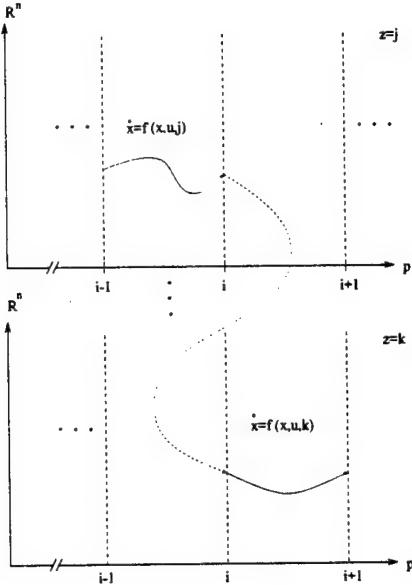
Figure 3-7: Example dynamics of Brockett's Type D model.

The first equation denotes the continuous dynamics and the last equation the "symbolic processing" done by the system. The times when p passes through integer values can be thought of as the discrete event times of the hybrid dynamical system. Thus, we consider BD as a precise, first-order model of interactions of ODEs and DEDS. Once again one may introduce output equations:

$$
\begin{aligned}
y(t) &= c(x(t), z\lfloor p \rfloor), \\
o\lfloor p \rfloor &= \eta(y[t], z\lfloor p \rfloor).
\end{aligned}
$$

Finally, Brockett generalizes BD to the case of "hybrid system with vector triggering" (herein, BDV), in which one replaces the single rate and symbolic equations with a finite number of such equations:

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), z\lfloor p \rfloor), \\
\dot{p}_i(t) &= r(x(t), u(t), z_i\lfloor p_i \rfloor), \\
z_i\lceil p_i \rceil &= \nu(x[t]_{p_i}, z\lfloor p \rfloor, v_i\lfloor p_i \rfloor),
\end{aligned}
$$

where $i \in \{1, 2, \ldots, k\}$. Again, outputs may be introduced.

In order for, say, the BB system above to be well-posed, we would like there to exist a unique solution on finite interval $[0, T]$. That is, given $[x(0), p(0)] \in X \times \mathbb{R}$, there should exist unique $x(\cdot)$ and $p(\cdot)$, continuous and differentiable almost everywhere, satisfying the equations. Brockett meets these specifications on any interval in which p does not take on an integral value by requiring f to be Lipschitz in x, continuous with respect to u, and u to have a finite number of discontinuities. [It is also necessary to assume U bounded]. The result extends if on any finite interval of time p passes through only a finite number of integers, leading to a finite number of discontinuities of the derivatives of x and p in finite time. In general, this requires similar continuity assumptions on r. Consider, for example, the case where $V = \mathbb{Z}_+$, $v\lfloor p \rfloor = (\lfloor p \rfloor + 1)^2$, and $r = v\lfloor p \rfloor$. This leads to $\dot{p} = (\lfloor p \rfloor + 1)^2$, which has finite escape time. Analogous behavior for x results if u is not bounded. In the usual case, however, U, V, and Z are taken compact, avoiding such behavior. Similar discussion holds for models BD and BDV.

In [38], Brockett gives many examples of systems modeled with the above equations, including buffers, stepper motors, and transmissions (cf. Example 3.4).

§3.9 DISCUSSION OF REVIEWED MODELS

At the risk of oversimplification, WHS, TDA, NKSD, and ASL use autonomous switching; BGM uses autonomous switching and autonomous jumps; and BD uses a combination of autonomous and controlled switching.

Also, comparing to systems with impulse effect [§2.2.3], we have the following. BGM can be modeled as a Class III system (cf. §5.2); TDA, and autonomous versions of WHS, NKSD, and ASL [§3.10] are special cases of Class III systems; the autonomous versions of BB and BD [§3.10] are Class II.

From the control perspective, the TDA model is an autonomous system and the BGM is essentially so (although one can set parameters on jumps). The NKSD and ASL models focus on the "control automaton," coding the action of the controller in the mappings from continuous states to input symbols, through automaton to output symbols, and back to controls.

Witsenhausen adds a control to the continuous component of the system dynamics. Brockett's BD/BDV models allow the possibility of both continuous and discrete controls to be exercised as input to the continuous and symbolic dynamics of the systems, respectively. That is the plant not only responds to the state (or output) of the finite machine, but to continuous commands generated separately as well. One may argue that this is largely a matter of level of modeling. For instance, one can assume (as in NKSD and ASL) that the "low-level" loops have been closed, eliminating the continuous control from the design of the "high-level" ones. Nevertheless, our approach in this thesis is more in spirit with those of Witsenhausen and Brockett.

From the original papers, it is clear that the models above were primarily developed for a variety of purposes: TDA and BGM for modeling and simulation, NKSD and ASL for controlling continuous systems with computer programs or "higher level controllers," and Brockett's for modeling the action of (hierarchical) motion control systems. Moreover, there is a direct trade-off between the generality of a model and what one can prove about such a model. Therefore, "containment" of one model in another does not reflect any bias of the more general model's being "superior." Indeed, in §5 we develop a very general, abstract model which captures many hybrid phenomena and the models reviewed here. Later, however, we place restrictions on this model in order to solve a related control

problem [§10].

§3.10   COMPARISON OF REVIEWED MODELS

In the sequel, we explore the capabilities of the hybrid systems models WHS, TDA, BGM, NKSD, ASL, and BD, described above. Clearly, these models were developed for different purposes with assumptions arising accordingly. Nevertheless—and for expediency—we note some containment relations among these models.

Here, A **contains** B means that every system described by the equations of model B can be described by the equations of model A. When the equations of a model describe a system, we say that the model **implements** that system.

First, since we are not interested in control yet, we develop autonomous versions of the models WHS, NKSD, ASL, and BD above, in which the control inputs are replaced by fixed functions of state. For instance, an autonomous version of WHS (denoted WAUT) arises by dropping dependence of the vector field on input $u(t)$.

Next, we construct an autonomous version of NKSD. An autonomous version of ASL (denoted ASLAUT) can be constructed similarly. We refer to the following as NKAUT.

$$\begin{aligned}
\dot{x}(t) &= f(x(t), q(t)), \\
q^+(t) &= \nu(q(t), AD(x(t), q(t))),
\end{aligned}$$

where $x(t) \in \mathbb{R}^n$ and $q(t) \in Q \simeq \{1, \ldots, N\}$. Here, $f : \mathbb{R}^n \times Q \to \mathbb{R}^n$, $\nu : Q \times I \to Q$, and $AD : \mathbb{R}^n \times Q \to I \simeq \{1, \ldots, M\}$. Note that we have incorporated the output equations into the f, ν, and AD functions. The AD map is restricted as discussed in §3.6.

Here is an autonomous version of the BD model, which we refer to as BAUT:

$$\begin{aligned}
\dot{x}(t) &= f(x(t), z\lfloor p \rfloor), \\
\dot{p}(t) &= r(x(t), z\lfloor p \rfloor), \\
z\lceil p \rceil &= \nu(x\lfloor t_p \rfloor, z\lfloor p \rfloor, \lfloor p \rfloor),
\end{aligned}$$

where $x(t) \in \mathbb{R}^n$, $p(t) \in \mathbb{R}$, $z\lfloor p \rfloor \in Z \simeq \{1, \ldots, N\}$, $f : \mathbb{R}^n \times Z \to \mathbb{R}^n$, $r : \mathbb{R}^n \times Z \to \mathbb{R}$, and $\nu : \mathbb{R}^n \times Z \times \mathbb{Z} \to Z$. As in BD, r is restricted to be nonnegative.

By construction, the original models contain their autonomous versions. Note also that BAUT is distinct from the TDA and NKAUT models since, for instance, it allows arbitrary dependence of the discrete dynamics ν on $x\lfloor t_p \rfloor$, which can lead to partitions not permitted by the other two models. We have other containment relations as follows.

Remark 3.6 *BGM contains TDA.*

Proof.   Given an arbitrary TDA equation, choose, in the BGM model, $X_q = \mathbb{R}^n$, $U_q = C(q) = \mathbb{R}^n \backslash M_q$, $f_q = f(\cdot, q)$, $G_q(x) = (x, p)$ if $x \in M_{q,p}$, and $h_{q,p} \equiv -g_{q,p}$ for all $q \in Q$, $p \in I(q)$. $\square$

Remark 3.7 *NKAUT contains TDA.*

Proof.   Suppose we are given an arbitrary TDA equation (i.e., a differential automaton A). Let primed symbols denote those in the NKAUT model with the same notation as those for the differential automaton. Set $Q' = Q$, $f'(\cdot, q) = f(\cdot, q)$, $q \in Q'$. This duplicates the continuous dynamics.

Now, for each $q \in Q'$, choose the small topology on $\mathbb{R}^n$

$$\mathcal{T}_q = C(q) \cup \bigcup_{p \in I(q)} M^\epsilon_{q,p},$$

where $0 < \epsilon < \alpha(A)/3$ and

$$M^\epsilon_{q,p} \equiv \{x \in \mathbb{R}^n | \operatorname{dist}(x, M_{q,p}) < \epsilon\}.$$

The non-empty join irreducibles are $C(q)$ and $M^\epsilon_{q,p}$, $A^\epsilon_{q,p}$, $p \in I(q)$, where

$$A^\epsilon_{q,p} \equiv M^\epsilon_{q,p} \backslash M_{q,p}.$$

Let $i_{q,p}$, $j_{q,p}$, k_q denote the symbols associated with the join irreducibles $M^\epsilon_{q,p}$, $A^\epsilon_{q,p}$, and $C(q)$, respectively. Defining

$$\begin{aligned}
\nu'(q, i_{q,p}) &= p, \\
\nu'(q, j_{q,p}) &= q, \\
\nu'(q, k_q) &= q,
\end{aligned}$$

duplicates the discrete dynamics. $\qquad\square$

Remark 3.8 *BGM contains BAUT.*

Proof. Given an arbitrary BAUT equation, choose, in the BGM model,

$$\begin{aligned}
X_1 &= \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}, \\
U_1 &= \mathbb{R}^n \times (-\infty, 1) \times \mathbb{R} \times \mathbb{R}, \\
f_1(x, q, p, z) &= [f(x, z), \, r(x, z), \, 0, \, 0], \\
G_1(x, q, p, z) &= (x, 0, p+1, \nu(x, z, p), 1), \\
h_1(x, q, p, z) &= 1 - q.
\end{aligned}$$

$\qquad\square$

Notice the last proof shows that in the BGM model, setting parameters on hitting switching boundaries can be implemented with the transition functions. Note also that unlike the first two proofs, the last construction uses a different (but equivalent) state space for BGM and BAUT. In any case, we do not use the fact that BGM contains BAUT in further results. Also, we do not compare among BGM, NKSD, and BD here. We also notice that WAUT and ASLAUT contain TDA.

Summarizing results needed later, the WHS, BGM, NKSD, and ASL models contain the TDA model; BD contains BAUT. Thus in the sequel, when examining capabilities of these models, the presentation concentrates on the TDA and BAUT models, since all capabilities possessed by both of them are automatically possessed by all those reviewed above. Extra capabilities of the BGM model are noted as warrants.

§3.11 NOTES

Our classification of hybrid phenomena and the review of all but Witsenhausen's model appeared in [32]. The comparison results for those five minus the ASL model is from [31].

Chapter 4

Classification of Hybrid Systems

In this chapter we classify hybrid systems according to their structure and the phenomena that they exhibit. The hierarchy of classes we explore are as follows (for both autonomous and controlled systems). First, there are *general hybrid dynamical systems* (*GHDS*). These are then refined to the concept of *hybrid dynamical system*, or simply *hybrid system*, studied in this thesis. Then there are two restrictions of hybrid systems, in which the discrete dynamics are suppressed and no continuous-state jumps are allowed: *switched systems* and *continuous switched systems*.

Along the way we also explicitly define the dynamics of general hybrid dynamical systems.

§4.1 GENERAL HYBRID DYNAMICAL SYSTEMS

Recall the definition of a general dynamical system (GHDS):

$$H = [Q, \mathbf{\Sigma}, \mathbf{A}, \mathbf{G}],$$

with its constituent parts defined in §1.4. Also recall the notation and shorthand from that section. We now offer the following observations, expanded from that section.

Dynamical Systems. First, note that in the case $|Q| = 1$ and $A = \emptyset$ we *recover all dynamical systems*. Thus, the class of GHDS includes systems described by ordinary differential equations, difference equations, and digital automata (which include finite automata, pushdown automata, Turing machines, Petri nets, etc.)

Hybrid Systems. Next, the case $|Q|$ finite, each X_q a subset of $\mathbb{R}^n$, and each $\Gamma_q = \mathbb{R}$ largely corresponds to the *usual* notion of a hybrid system, viz. a coupling of finite automata and differential equations [31, 32, 66]. The two are coupled at "event times" when the continuous state hits certain boundaries, prescribed by the sets A_q. We thus include the examples containing the discrete phenomena discussed in §3.2: collisions, relays, hysteresis, etc. This is proven more formally in §5, where we examine the following useful extension, repeated from §1.4:

> A **hybrid system** is a general hybrid dynamical system with Q countable, and with $\Gamma_q \equiv \mathbb{R}$ (or $\mathbb{R}_+$) and $X_q \subset \mathbb{R}^{d_q}$, $d_q \in \mathbb{R}_+$, for all $q \in Q$. In the notation above, it may be written as
>
> $$[Q, [\{X_q\}_{q \in Q}, \mathbb{R}_+, \{f_q\}_{q \in Q}], \mathbf{A}, \mathbf{G}],$$

where f_q is a vector field on $X_q \subset \mathbb{R}^{d_q}$.

Here, we may take the view that the system evolves on the state space $\mathbb{R}^* \times Q$, where $\mathbb{R}^*$ denotes the set of finite, but variable-length real-valued vectors. For example, Q may be the set of labels of a computer program and $x \in \mathbb{R}^*$ the values of all currently-allocated variables. If $N = \sup\{d_q\}_{q \in Q}$ is finite, we may take the state to belong to the so-called carrier manifold $\mathbb{R}^N$.

> EXAMPLE. An interesting example motivated above is scientific calculations. More precisely, we can model Smale's tame machines [19] in this framework.

Changing State Space.   The state space may change. This is useful in modeling component failures or changes in dynamical description based on autonomous—and later, controlled—events which change it. Examples include the collision of two inelastic particles or an aircraft mode transition that changes variables to be controlled [103].

We allow the X_q to overlap and the inclusion of multiple copies of the same space. This may be used, for example, to take into account overlapping local coordinate systems on a manifold [9].

Refinements.   We may refine the concept of general hybrid dynamical system H by adding:

- inputs, including control inputs, disturbances, or parameters (see general *controlled* hybrid dynamical system below).

- outputs, including **state-output** for each constituent system as for dynamical systems [§2.1.2] and **edge-output**:

$$H = [Q, \Sigma, \mathbf{A}, \mathbf{G}, \mathbf{O}, \eta] ,$$

   where $\eta : A \to O$ produces an output at each jump time.

- $I, F \subset S$, sets of initial or final states.

- $\Delta : A \to \mathbb{R}_+$, the **jump delay map**, which can be used to account for the time which abstracted-away, lower-level transition dynamics actually take.

   > EXAMPLE. Think of modeling the closure time of a discretely-controlled hydraulic valve or trade mechanism imperfections in economic markets.

- Marked states, timing, or input and output for any constituent system.

Example 4.1 (Reconciling Time Scales) *Suppose that each constituent dynamical system Σ_q of H is equipped with a timing map. That is $\tau = \{\tau_q\}_{q \in Q}$ where*

$$\tau_q : X_q \times \Gamma_q \to \mathbb{R}_+ .$$

Then, we may construct **trajectories** *for H, i.e., a function from "real-time" to state. This is discussed in more generality in §4.3 below.*

Hierarchies.   We may iteratively combine hybrid systems H_q in the same manner, yielding a powerful model for describing the behavior of hierarchical systems (cf. Baas's hyperstructures [8] and Harel's statecharts [69]).

Adding Control.   Likewise, a general controlled hybrid dynamical system (GCHDS) is a system

$$H_c = [Q, \Sigma, \mathbf{A}, \mathbf{G}, \mathbf{V}, \mathbf{C}, \mathbf{F}],$$

with its constituent parts defined in §1.4.

The **admissible control actions** available are

- the **continous controls** $u \in U_q$, exercised in each constituent regime,

- the **discrete controls** $v \in V_q$, exercised at the autonomous jump times (which occur on hitting the set A),

- the **intervention times** and **destinations** of the controlled jumps.

NOTES.

1. Disturbances and other nondeterminism may be modeled by partitioning U, V, and C into portions that are under the influence of the controller or nature respectively.

2. The model includes that posed by Branicky, Borkar, and Mitter [32] and thus several other previously posed hybrid systems models [5, 9, 38, 114, 135, 152, §3.1]. It also includes systems with impulse effect [10, §2.2.3] and hybrid automata [57, §2.2.6]. See §5.2.

3. We could, but do not for pedagogical reasons and later results, combine autonomous and controlled jumps by defining a set-valued *autonomous* jump map $G' : A \cup C \to 2^S$ by

$$G'(s) = \begin{cases} G(s, V), & s \in A, \\ F(s) \cup \{s\}, & s \in C. \end{cases}$$

§4.2 CLASSIFYING GHDS

The scope of hybrid dynamical systems presents a myriad of modeling choices. In this section, we classify them according to their structure and the discrete phenomena they possess. Below, the prefixes "c-," "d-," and "t-" are used as abbreviations for "continuous-," "discrete-," and "time-" respectively. If no prefix is given, either can be used.

NOTE. To respect the historical development of the subject of hybrid systems and the cases of current high interest, we colloquially use "discrete" when referring to the index set, "continuous" when referring to the constituent state spaces, and "time" when referring to the constituent transition semigroups. We use these even though our model allows, for instance, the index state to be a continuum and the constituent state spaces to be discrete topological spaces.

Our *structural classification* is roughly captured by the following list.

- **Time-uniform.** The semigroups may be all be the same for each q, denoted $\Sigma = [\{X_q\}_{q \in Q}, \Gamma, \{\phi_q\}_{q \in Q}]$.

- **Continuous-time, discrete-time, sampled-data.** Each constituent dynamical system may be of a special type that evolves in continuous-time ($\Gamma = \mathbb{R}$), discrete-time ($\Gamma = \mathbb{Z}$), or a mixture. However, if the GHDS is time-uniform, we refer to it by the appropriate label, e.g., continuous-time-uniform.

- **C-uniform.** The ambient state space may be the same for each q, that is,

$$\Sigma = [X, \{\Gamma_q\}_{q \in Q}, \{\phi_q\}_{q \in Q}].$$

- **C-Euclidean, c-manifold.** Each ambient state space may be Euclidean (subset of $\mathbb{R}^n$ in the usual topology) or a smooth manifold.

- **D-compact, d-countable, d-finite, n-state.** Special cases arise when the index space is compact, finite, or countably infinite. If $|Q| = n$, we say the GHDS is n-state.

- **D-concurrent** versus **d-serial.** We may or may not allow more than one discrete jump to occur at a given moment of time. This is related to the next classification.

- **Dynamically-uniform.** The dynamics may be the same for each q. Strictly, such a case would also require that the system be c-uniform and time-uniform:

$$\Sigma = [\{X_q\}_{q \in Q}, \{\Gamma_q\}_{q \in Q}, \phi] \quad (\text{or } \Sigma = [\{X_q\}_{q \in Q}, \{\Gamma_q\}_{q \in Q}, f]), \qquad \text{for all } q \in Q.$$

In these systems, then, the interesting dynamics arises from the transition map G_q. Such is the case with timed automata (see §2.2.6 for references).

- **Deterministic** versus **nondeterministic.**

- **Nonautonomous** versus **autonomous.** The continuous (or discrete) portions of the dynamics may or may not depend on time (or count of events) or external controls. To distinguish between time and control dependence, we use the next two classifications.

- **Time-varying** versus **time-invariant.**

  NOTE. We do not explicitly deal with time-varying systems here, assuming it is taken care of in the usual way, viz. appending another state to represent time.

- **Controlled** versus **uncontrolled.**

Finally, a hybrid dynamical system may also be classified according to the discrete dynamic phenomena that it exhibits as follows (cf. [32]).

- **Autonomous-switching.** The autonomous jump map $G \equiv \nu$ is the identity in its continuous component, i.e., $\nu : A \to S$ has $\nu(x, q) = (x, q')$.

- **Autonomous-impulse.** $G \equiv J$ is the identity in its discrete component.

- **Controlled-switching.** The controlled jump map F is the identity in its continuous component, i.e., $F(x, q) \subset \{x\} \times Q$.

- **Controlled-impulse.** F is the identity in its discrete component.

With this notation, our GHDS model admits some special cases:

- H autonomous-impulse with $|Q| = 1$ and $\Gamma = \mathbb{R}$ is an **autonomous system with impulse effect** [10].

- H c-uniform, time-uniform, and autonomous-switching is an **autonomous switched system** [30].

- H continuous-time-uniform, c-Euclidean-uniform, d-countable, is what we call a **hybrid system**.

§4.3 GHDS DYNAMICS

In this section, we place some restrictions on GHDS in order to prove some behavioral properties. We assume that Γ is an ordered set with the least upper bound property, equipped with the order topology. Note that this implies Γ is a lattice [88]. We also assume addition to be order-preserving in the sense that if $a > 0$, then $a + b > b$. This last assumption ensures, among other things, that

$$\Gamma^+ = \{a \in \Gamma \mid a \geq 0\}$$

is a semigroup; likewise for Γ^-, defined symmetrically. For brevity, we call such a group (semigroup) **time-like**.

> EXAMPLE. The most widely used time-like groups are $\mathbb{R}$ and $\mathbb{Z}$ under addition, each in the usual order. The rationals are not allowed since they do not have the least upper bound property (though they are a lattice). However, $\rho\mathbb{Z}$, $\rho \in \mathbb{R}$, is allowed. Example semigroups are $\mathbb{R}_+$, $\rho\mathbb{Z}_+$, and the free monoid generated by a finite, ordered alphabet (in the dictionary order).

In this section we consider several initial value problems for GHDS. First, in the time-like case, given dynamical system $[X, \Gamma, \phi]$, we may define the **positive orbit** of the point x as $P(x) \equiv \phi(x, \Gamma^+)$.

> NOTE. The negative orbit $B(x)$ may be defined even in the non-reversible case by $y \in B(x)$ if and only if $x \in P(y)$ [17].

Problem 4.2 (Reachability Problem) *Compute the positive orbit for general hybrid dynamical system H.*

The positive orbit is the *set* defined as follows. We consider only initial points in $I \equiv X \backslash A$. We restrict ourselves to the case where A_q is closed[1] and $D_q \cap A_q = \emptyset$. Suppose $s_0 = (x_0, q_0) \in I$. If $P_{q_0}(x_0) = \phi_{q_0}(x_0, \Gamma^+)$ does not intersect A_{q_0} we are done: the positive orbit is just $P_{q_0}(x_0)$. Else, let

$$g_1 = \inf\{g \in \Gamma_{q_0}^+ \mid \phi_{q_0}(x_0, g) \in A_{q_0}\}. \tag{4.1}$$

---

[1]We may relax this to $\partial^* A_q$, defined in Equation (2.4), trivially.

Since A_{q_0} closed, Γ^+ is time-like, and ϕ is continuous, the set in Equation (4.1) is closed, g_1 exists, and $x_1^- = \phi_{q_0}(x_0, g_1) \in A_{q_0}$. Define $s_1 = G_{q_0}(x_1^-) \in I$ and continue. $\square$

When a GHDS is time-uniform with time-like group Γ, it induces a Γ^+-transition system $[I, \Gamma^+, \Phi]$. In this case, we may define its (forward) **trajectory** as a *function* from Γ^+ into I.

Problem 4.3 (Trajectory Problem) *Compute the trajectories for general hybrid dynamical system H.*

We want to compute $\Phi(s_0, g_f)$ for $g_f \in \Gamma^+$ and $s_0 = (x_0, q_0) \in I$. Let $g_0 = 0$. The construction and arguments are similar to above, so we give the $(i-1)$th step:

$$g_{i+1} = \inf\{g \in \Gamma_i^+ \mid g \leq g_f - g_i, \ \phi_{q_i}(x_i, g) \in A_{q_i}\}. \tag{4.2}$$

If the set is ever empty, we are done. On the interval $g \in [g_i, g_{i+1})$ the system evolves according to $\phi(x_i, g - g_i)$. Also, $g_{i+1} - g_i > 0$ is assured by the fact that topological groups are regular [113, p. 145]. $\square$

The above construction allows us to formulate stability and *finite-time reachability* problems.

Note that trajectories *may not be extendible to all of* Γ^+, i.e., we have not precluded the **accumulation** of an infinite number of jumps in finite time. See Example 10.7. This can be removed in the case of hybrid systems by, for example, assuming uniform Lipschitz continuity of the vector fields and uniform separation of the jump and destination set A and D. See §10.

NOTE. From above, if $D_q \cap A_q = \emptyset$ and A_q closed then (positive) orbits and trajectories exist (up to a possible accumulation time of finite jumps) and are unique. Using [17, Theorem 3.4.11], it is enough that $\Lambda_q \equiv D_q \cup (X_q \backslash A_q)$ be locally compact at each $x \in D_q$ and $f(x)$ subtangential to Λ_q for all $x \in D_q \cap \partial D_q$.

Similarly to reachability, we have the following.

Problem 4.4 (Accessiblity Problem) *Compute the set of points accessible under all admissible control actions from initial set I for general controlled hybrid dynamical system system H_c.*

The answer is largely the same as above except that we must vary over all admissible control actions.

§4.4 HYBRID DYNAMICAL SYSTEMS

In this section, we give explicit representations of the different classes of hybrid systems arising from the definitions above. We concentrate on the c-continuous-time, c-uniform, d-finite, time-invariant, autonomous case. Extensions to other cases above are straightforward.

NOTE. Recall the notation q^+, introduced in §3.2.5.

A (continuous-time) **autonomous-switching hybrid system** may be defined as follows:

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), q(t)), \\
q^+(t) &= \nu(x(t), q(t)),
\end{aligned}
\tag{4.3}
$$

where $x(t) \in \mathbb{R}^n$, $q(t) \in Q \simeq \{1, \dots, N\}$. Here, $f(\cdot, q) : \mathbb{R}^n \to \mathbb{R}^n$, $q \in Q$, each globally Lipschitz continuous, is the **continuous dynamics** of Equation (4.3); and $\nu : \mathbb{R}^n \times Q \to Q$ is the **finite dynamics** of Equation (4.3).

Thus, starting at $[x_0, i]$, the continuous state trajectory $x(\cdot)$ evolves according to $\dot{x} = f(x, i)$. If $x(\cdot)$ hits some $(\nu(\cdot, i))^{-1}(j)$ at time t_1, then the state becomes $[x(t_1), j]$, from which the process continues.

Clearly, this is an instantiation of autonomous switching. Switchings that are a fixed function of time may be taken care of by adding another state dimension, as usual. Examples are the Tavernini and autonomous Witsenhausen models.

By a **c-controlled autonomous-switching hybrid system** we have in mind a system of the form:

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), q(t), u(t)), \\
q^+(t) &= \nu(x(t), q(t), u(t)),
\end{aligned}
\tag{4.4}
$$

where everything is as above except that $u(t) \in \mathbb{R}^m$, with f and ν modified appropriately. An example is Witsenhausen's model.

An is a system

$$
\begin{aligned}
\dot{x}(t) &= f(x(t)), & x(t) \notin M \\
x^+(t) &= J(x(t)), & x(t) \in M
\end{aligned}
\tag{4.5}
$$

where $x(t) \in \mathbb{R}^n$, and $J : \mathbb{R}^n \to \mathbb{R}^n$. Examples include autonomous systems with impulse effect.

Finally, a hybrid system with autonomous switching and autonomous impulses (i.e., the full power of autonomous jumps) is just a combination of those discussed above:

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), q(t)), \\
x^+(t) &= J(x(t)), \\
q^+(t) &= \nu(x(t), q(t)),
\end{aligned}
$$

where $x(t) \in \mathbb{R}^n$ and $q(t) \in Q \subset \mathbb{Z}$. Examples include the BGM model and hence all the other autonomous models in §3.

Likewise, we can define discrete-time autonomous and controlled hybrid systems by replacing the ODEs above with difference equations. In this case, Equation (4.3) represents a simplified view of some of the models in [38]. Also, adding controls—both discrete and continuous—is straightforward. Finally, non-uniform continuous state spaces, i.e., $x(t) \in X_{q(t)}$, may be added with little change in the foregoing.

§4.5 SWITCHED SYSTEMS

We have in mind the following model as a prototypical example of a **switched system**:

$$
\dot{x}(t) = f_i(x(t)), \qquad i \in Q \simeq \{1, \dots, N\},
\tag{4.6}
$$

where $x(t) \in \mathbb{R}^n$. We add the following *switching rules*.

- Each f_i is globally Lipschitz continuous.

- The i's are picked in such a way that there are finite switches in finite time.

Switched systems are of "variable structure" or "multi-modal"; they are a simple model of (the continuous portion) of hybrid systems. The particular i at any given time may be chosen by some "higher process," such as a controller, computer, or human operator, in which case we say that the system is **controlled**. It may also be a function of time or state or both, in which case we say that the system is **autonomous**. In the latter case, we may really just arrive at a single (albeit complicated) nonlinear, time-varying equation. However, one might gain some leverage in the analysis of such systems by considering them to be amalgams of simpler systems.

Models like Equation (4.6) have been studied for stability [60, 119]. However, those papers were predominantly concerned with the case where all the f_i are linear. We discuss the general cases in §8.

We also discuss difference equations

$$x[k+1] = f_i(x[k+1]), \qquad i \in Q \simeq \{1, \ldots, N\}, \tag{4.7}$$

where $x[k] \in \mathbb{R}^n$. Here, we only add the assumption that each f_i is globally Lipschitz continuous. Again, these equations can be thought of as the "continuous" portion of the dynamics of hybrid systems combining difference equations and finite automata [38].

More abstractly, a **general switched system** is a system

$$\sigma = [Q, \Sigma, \mathbf{A}, \mathbf{R}],$$

where all is as defined as above, except the operators A_q, which are restricted to be *autonomous-switching*, and the **switching rules** given by $\mathbf{R}$. We will not be more precise here, noting that the cases of usual interest (i.e., when $\mathbf{R}$ depends on the state) have been taken care of above. Nevertheless, the abstraction is helpful, as seen below, where we concentrate on the *non-general* case.

Our abstracting away of the finite dynamics in studying switched systems above can be motivated by "verification by successive approximation" [2]. For instance, consider the following set of (initial state-dependent) switching sequences:

(C) Constrained (hybrid systems)

(A) Arbitrary (IFS, see §8)

Then, examining the reachability sets of these systems with respect to safety and liveness constraints given by sets, we have the following picture:

- Safety: $\mathcal{R}(C) \subset \mathcal{R}(A) \subset$ **Safe**

- Liveness: **Live** $\subset \mathcal{R}(C) \subset \mathcal{R}(A)$

Such containment relations allow one to prove properties of the hybrid system by comparison with the switched system and vice versa.

§4.6 CONTINUOUS SWITCHED SYSTEMS

We also study *continuous* switching systems. A **continuous switching system** is a switching system with the additional constraint that the switched subsystems agree at

the switching time. More specifically, consider Equation (4.6) and suppose that at times t_j, $j = 1, 2, 3, \ldots$, there is a switch from $f_{k_{j-1}}$ to f_{k_j}. Then we require $f_{k_{j-1}}(x(t_j), t_j) = f_{k_j}(x(t_j), t_j)$. That is, we require that the vector field is continuous over time.

This constraint leads to a simpler class of systems to consider. At the same time, it is not overly restrictive since many switching systems naturally satisfy this constraint. Indeed they may even arise from the discontinuous logic present in hybrid systems. For example, we might have an aircraft where some surface, say the elevator, controls the motion. But this elevator is in turn a controlled surface, whose desired action is chosen by a digital computer that makes some logical decisions. Based on these decisions, the computer changes elevator inputs (say current to its motor) in an effectively discontinuous manner. However, the elevator angle and angular velocity do not change discontinuously. Thus, from the aircraft's point of view (namely, at the level of dynamics relevant to it), there are continuous switchings among regimes of elevator behavior. Therefore, continuous switching systems arise naturally from abstract hybrid systems acting on real objects.

Another problem arises in examples like the one we just introduced: the problem of unmodeled dynamics. Suppose the pilot, from some quiescent operating point, decides to invoke hard upward normal acceleration. The elevator starts swinging upward until it is swinging upward at maximum angular velocity (in an effort track the violent maneuver requested by the pilot). Then, some higher process makes a calculation and decides that continuing this command would result in an unsafe configuration (say attack angle beyond critical). It decides to begin braking the elevator motor immediately to avoid this situation. In this case, the *desired* angular velocity profile of the elevator (over the whole move) is most probably trapezoidal. However, the elevator is a dynamic element that can't track that desired profile exactly. We may want to know how taking these unmodeled dynamics into account affects our already modeled dynamics. We may also want to know how high our control gains should be to track within a certain error. In §8 we develop theory that allows to answer both these questions.

§4.7 NOTES

Our identification of discrete phenomenon and examples is from [32]. We started to classify hybrid systems in [30]. That paper also studied switched systems. Continuous switched systems were pursued in [21], summarized in [28]

Chapter 5

Unified Hybrid Systems Model

In this chapter we introduce an abstract, unified hybrid systems model which is shown to capture all identified discrete phenomena arising in hybrid systems and subsume all reviewed and classified hybrid systems models. The resulting model is useful for posing and solving hybrid analysis and control problems.

§5.1 OUR UNIFIED MODEL

We now present our over-riding hybrid systems framework in generality. We refine it later when we set up our control problem in §10. In the nomenclature of §4, the result is a *controlled hybrid system with delay maps*. For simplicity, though, the destination sets are specified *a priori* instead of by the collection of set-valued maps **F**. Also, with no real loss of generality, we consider $U_i \equiv U$, $V_i \equiv V$, $i \in \mathbb{Z}_+$. However, we do generalize to allow the vector field to depend on the continuous state at the last jump time.

Specifically, our discrete state space is $Q = \mathbb{Z}_+$. The continuous state space for $x(\cdot)$ is $X = \{X_i\}_{i=0}^{\infty}$ where each X_i is a subset of some Euclidean space $\mathbb{R}^{d_i}$, $d_i \in \mathbb{Z}_+$. We also specify *a priori* regions $A_i, C_i, D_i \subset X_i$, $i \in \mathbb{Z}_+$. These are the **autonomous jump sets**, **controlled jump sets**, and **jump destination sets**, respectively. Let A, C, and D denote the unions $\bigcup_i A_i \times \{i\}$, $\bigcup_i C_i \times \{i\}$, and $\bigcup_i D_i \times \{i\}$, $i \in \mathbb{Z}_+$, respectively. Let U, V be the sets of continuous and discrete controls, respectively. The following maps are assumed to be known:

1. **vector fields** $f_i : X_i \times X_i \times U \to \mathbb{R}^{d_i}$, $i \in \mathbb{Z}_+$.

2. **jump transition maps** $G_i : A_i \times V \to D$.

3. **autonomous transition delay** $\Delta_{a,i} : A_i \times V \to \mathbb{R}_+$.

4. **controlled transition delay** $\Delta_{c,i} : C_i \times D \to \mathbb{R}_+$.

As shorthand, we may define $G : A \times V \to D$ in the obvious manner. Similarly, for Δ_a and Δ_c.

The dynamics of the control system can now be described as follows. There is a sequence of **pre-jump times** $\{\tau_i\}$ and another sequence of **post-jump times** $\{\Gamma_i\}$ satisfying $0 = \Gamma_0 \le \tau_1 < \Gamma_1 < \tau_2 < \Gamma_2 < \cdots \le \infty$, such that on each interval $[\Gamma_{j-1}, \tau_j)$ with non-empty interior, $x(\cdot)$ evolves according to the differential equation

$$\dot{x}(t) = \xi(t), \qquad t \ge 0$$

79

in some X_i, $i \in \mathbb{Z}_+$. At the next pre-jump time (say, τ_j) it jumps to some $D_k \in X_k$ according to one of the following two possibilities:

1. $x(\tau_j) \in A_i$, in which case it *must* jump to $x(\Gamma_j) = G_i(x(\tau_j), v_j) \in D$ at time $\Gamma_j = \tau_j + \Delta_{a,i}(x(\tau_j), v_j)$, $v_j \in V$ being a control input. We call this phenomenon an autonomous jump.

2. $x(\tau_j) \in C_i$ and the controller *chooses* to—it does not have to—move the trajectory discontinuously to $x(\Gamma_j) \in D$ at time $\Gamma_j = \tau_j + \Delta_{c,i}(x(\tau_j), x(\Gamma_j))$. We call this a controlled (or impulsive) jump.

See Figure 5-1.

Figure 5-1: Example dynamics of our unified model.

For $t \in [0, \infty)$, let $[t] = \max_j\{\Gamma_j \mid \Gamma_j \leq t\}$. The vector field $\xi(t)$ of Equation (3.1) is given by

$$\xi(t) = f_i(x(t), x[t], u(t)), \tag{5.1}$$

where i is such that $x(t), x[t] \in X_i$ and $u(\cdot)$ is a U-valued control process.

NOTE. The autonomous version of this model (including no controlled jumps) yields unique trajectories in the case of, for instance, A_i closed and $A_i \cap D_i = \emptyset$. See §4.

§5.2 INCLUSION OF DISCRETE PHENOMENA AND PREVIOUS MODELS

We now show how this framework encompasses the discrete phenomenon of §3.2, and how it subsumes the hybrid systems models reviewed in §3 and classified in §4.

First, a simplification. If a set of parameters or controls is countable and discrete, such as a set of strings, we may take it to be isomorphic with a subset of $\mathbb{Z}_+$. On the other hand, consider a set of parameters or controls, U, where U is a compact, connected, locally connected metric space U. By the Hahn-Mazurkiewicz theorem [75], U is the continuous image of $[0, 1]$ under some map and thus we may set $U = [0, 1]$ without any loss of generality. Thus, we may assume below without any loss of generality that parameters and controls take values in a subset $P \subset \mathbb{R}^m$.

Autonomous Switching. We show that autonomous switching can be viewed as a special case of autonomous jumps, which are taken care of next. Consider the differential equation with parameters

$$\dot{x} = f(x, p),$$

where $x \in \mathbb{R}^n$, $p \in P \subset \mathbb{R}^m$ closed, and $f : \mathbb{R}^n \times P \to \mathbb{R}^n$ continuous. Let, $\nu : \mathbb{R}^n \times P \to P$ be the function governing autonomous switching. For example, in the Tavernini model, ν is the "discrete dynamics."

Then, since $\mathbb{R}^n$ has the universal extension property [113, §A], we can extend f to a continuous function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Now, consider the ODE on $\mathbb{R}^{n+m}$:

$$\begin{aligned} \dot{x} &= F(x, \xi), \\ \dot{\xi} &= 0, \end{aligned}$$

where $x \in \mathbb{R}^n$, $\xi \in \mathbb{R}^m$, and $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ continuous. Let, the transition function be $G : \mathbb{R}^n \times P \to \mathbb{R}^n \times P$ with $G(x, p) = (x, \nu(x, p))$.

Autonomous Jumps. This is clearly taken care of with the sets A_i.

Controlled Switching. A system with controlled switching is described by

$$\dot{x}(t) = f(x(t), u(t)), \qquad x(0) = x_0 \in \mathbb{R}^d,$$

where $u(\cdot)$ is a piecewise constant function taking values in $U \subset \mathbb{R}^m$ and $f : \mathbb{R}^d \times U \to \mathbb{R}^d$ is a map with sufficient regularity. There is a strictly positive cost associated with the switchings of $u(\cdot)$. In our framework, let $x'(\cdot) = [x(\cdot), \ u(\cdot)]^T$ be the new state process with dynamics

$$\dot{x}'(t) = f'(x'(t)), \qquad f'(\cdot) = [f(\cdot), \ 0]^T,$$

taking values in $X = \{X_i\}_{i=0}^{\infty}$ where each X_i is a copy of $\mathbb{R}^d \times U$. Set $C_i = D_i = X_i$, $A_i = \emptyset$ for $i \in \mathbb{Z}_+$. Switchings of $u(\cdot)$ now correspond to controlled jumps with the associated costs.

Controlled Jumps. This is clearly taken care of with the sets C_i.

Digital Automata. A variety of automata are automatically subsumed by inclusion of the Tavernini, BGM, NKSD, ASL, and Brockett models, which is demonstrated next. Inclusion

of general digital automata follows by using a countable symbol set I in, for instance, the NKSD model.

Tavernini and BGM Models. Let primed symbols denote those in our model with the same notation as those for BGM. It is obvious that our model includes the BGM model by choosing $X_i' = \overline{X_i}$, $A_i = \partial X_i \cup \partial U_i$, $D_i = \overline{U_i}$, $C_i = \emptyset$, and

$$G_j'(x) = (\pi_1(G_j(x)); \ \pi_2(G_j(x)))$$

for $x \in U_j$. G_j' need not be defined on $\partial X_j \backslash \partial U_j$, but for completeness we may define $G_j'(x) = (x; j)$ for $x \in \partial X_j \backslash \partial U_j$. Since BGM contains Tavernini's model, our model does as well.

NKSD and ASL Models. Our model includes the ASL model. First, choose $X_i = \mathbb{R}^n \times \mathbb{R}^3$, $i \in I$. Then, note that the sets $AD^{-1}(i)$, $i \in I$, form a partition of Y. Define the sets $M_j = h^{-1}(AD^{-1}(j))$ and define

$$A_i \equiv \bigcup_{j \neq i, j \in I} M_j.$$

Then define

$$\tilde{f}_i = [f, \ 0, \ 0, \ 0]$$

with dimensions representing x, q, i, and o. The model is complete by specifying

$$G_i(x) = (x, \nu(q, j), j, \eta(q); j)$$

if $x \in M_j \subset A_i$.

Inclusion of NKSD is similar. However, since the resulting partitions depend on q one must use multiple copies of X_i and $\tilde{f}_i$ as above, one for each AD_q, $q \in Q$. We must also append the state t to the state vector X (and use $[t]$), with the obvious differential equations/transitions. Finally, η depends on both q and j in this case.

Brockett's Models. Our model includes Brockett's BD model by choosing $X_1 = \mathbb{R}^n \times \mathbb{R}^4$ and defining

$$\tilde{f} = [f, \ r, \ 0, \ 0, \ 0]$$

with dimensions representing x, $q = p - \lfloor p \rfloor$, $i = \lfloor p \rfloor$, v, and z. Also, set $A_1 = \mathbb{R}^n \times \{1\} \times \mathbb{R}^3$, $D_1 = \mathbb{R}^n \times \{0\} \times \mathbb{Z}_+^3$, and $G_1((x, 1, i, v, z), v') = (x, 0, i + 1, v', \nu(x, z, v); 1)$. BB is seen to be included in the same manner, but removing the state dimension for z. It is clear that this can be extended to include BDV.

Setting Parameters and Timers. A system which, upon hitting boundaries, sets parameters from an arbitrary compact set $P \subset \mathbb{R}^p$ can be modeled in our framework by redefining $X_i' = X_i \times \mathbb{R}^p$, and $V' = V \times P$, and defining $f_i' : X_i \times X_i \times U \to \mathbb{R}^{d_i} \times \mathbb{R}^p$ as

$$f_i'(x, p, y, q, u) = [f_i(x, y, u), \ 0]^T$$

and $G' : A \times P \times V \times P \to D \times P$ as

$$G'(x, p, v, p') = [G(x, v), \ p']^T,$$

each for all possible arguments. Likewise, one can redefine the switching cost and delay appropriately.

A system which sets timers upon hitting boundaries can be modeled by a vector of the rate equations in Brockett's BDV model of hybrid systems, which in turn can be modeled in our framework as previously discussed.

§5.3 EXAMPLE

Consider again the hysteresis example of Equation (1.1). For specificity, consider a system with control, namely, $f \equiv H(x) + u$. It can be modeled as follows. The state space is $X = \{X_{-1}, X_1\}$, with $X_{-1} = [-\Delta, \infty)$ and $X_1 = (-\infty, \Delta]$. The continuous dynamics is given by

$$
\begin{aligned}
f_{-1} &= u - 1, \\
f_1 &= u + 1.
\end{aligned}
$$

The discrete dynamics is governed by the autonomous jump sets A_{-1} and A_1 and their associated transitions, which are, respectively,

$$
\begin{aligned}
(-\Delta, -1) &\mapsto (-\Delta, +1), \\
(+\Delta, +1) &\mapsto (+\Delta, -1).
\end{aligned}
$$

§5.4 NOTES

Our unified model first appeared in [32].

Part II

Analysis of Hybrid Systems

Chapter 6

Topology of Hybrid Systems

In this chapter, we discuss topological issues associated with hybrid systems. Recall from §1.6 that, in general, hybrid systems do not give rise to trajectories that are continuous in the initial condition. Therefore, the best one can hope for, in general, is continuity of each of the constituent maps. By construction though, each extended transition map, ϕ_q, is continuous. In this chapter, we discuss topologies such that the discrete component of the transition map, $\eta(\cdot, q)$, is continuous for each q. In particular, we examine topologies for achieving continuity of maps from a set of measurements of continuous dynamics to a finite set of input symbols (AD map).

Then we look at "completing the loop" by composing the AD map with that from a finite set of output symbols back into the control space for the continuous dynamics (DA map). Finding some anomalies in completing this loop, we discuss a different view of hybrid systems that can broach them and is more in line with traditional control systems. The most widely used fuzzy control system is related to this different view and does not possess these anomalies. Indeed, we show that fuzzy control leads to continuous maps (from measurements to controls) and that all such continuous maps may be implemented via fuzzy control.

We end by drawing connections to the previous and next chapters.

§6.1 INTRODUCTION

In traditional feedback control systems—continuous-time, discrete-time, sampled-data—the maps from output measurements to control inputs are continuous (in the usual metric-based topologies). When dealing with hybrid systems, however, one immediately runs into problems with continuity using the "usual" topologies. Whereby we begin ...

In this chapter, we discuss some results relating to the topology of hybrid (mixed continuous and finite dynamics) systems. We begin with a model of a hybrid system as shown in Figure 6-1.

We are interested in maps from the continuous plant's output or measurement space into a finite set of symbols. We call these AD maps. We are also interested in the map from this symbol space into the control or input space of the continuous plant (DA map). In many control applications, both the measurement and control spaces are (connected) metric spaces. Therefore, we keep our discussion germane to such assumptions.

The chapter is organized as follows: In the next section, we discuss AD maps. First, we illuminate the general issues. Then, we examine at length an AD map proposed in [114]. We verify that the map is indeed continuous, developing enough technical lemmas to easily add the fact that the symbol space topology they constructed is the same as the quotient
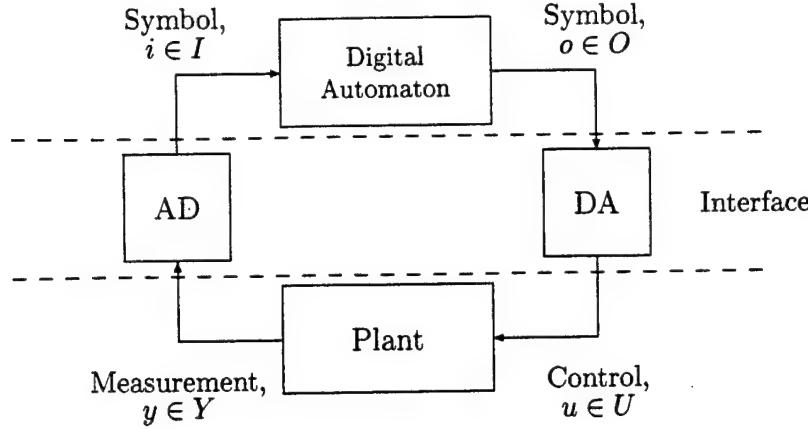
Figure 6-1: Prototypical hybrid system.

topology induced by their AD map.

In §6.3 we discuss what happens if we try to impose continuity from the measurement to the control spaces. We first illuminate why this is unreasonable given the fact that the measurement and control spaces are normally connected metric spaces. We then impose a new topology on the control space that gives rise to continuous maps.

In §6.5 we introduce a different view of hybrid systems. This view allows us to meaningfully discuss continuity of maps from the measurement to control spaces without introducing new topologies. We also show that the most widely used fuzzy logic control structure is related to this form, and that it indeed is a continuous map from measurements to controls. It is further demonstrated that these fuzzy logic controllers are dense in the set of such continuous functions.

We end with a topological viewpoint that reconciles the results in this chapter to our discussion of hybrid system trajectories in the preceding one and our definitions of simulation in the next. simulation, our next topic. The Appendix to this chapter collects the proofs of parts of a technical lemma. The thesis Appendix §A reviews most of the topological concepts used.

§6.2 CONTINUOUS AD MAPS

§6.2.1 GENERAL DISCUSSION

In this section, we discuss continuity of maps from the measurement space of the continuous plant into the finite symbol space. Such continuity is desirable when implementing control loops, since we want, roughly, small changes in measurement to lead to small changes in control action.

The basic problem we have in going from the continuum, Y, into a finite set of symbols, I, is that I usually comes equipped with the discrete topology and the only continuous maps from Y to I in this case are constant (since Y is connected and any subset of I with more than one point is not). Therefore, we must search for topologies on I which are not the discrete topology. At first, we may be disheartened by the fact that this also precludes all Hausdorff, and even T_1 topologies, from consideration. However, the topologies associated with (finite) observations are naturally T_0 [141]. Fortunately, there do exist T_0 topologies other than the discrete topology on any finite set of more than one point:

Example 6.1 *Suppose X is a finite set having $n > 1$ elements. There exists a T_0 topology on X that is not the discrete topology (and hence neither T_1 nor Hausdorff).*

Proof. Take as a basis the following subsets of X: $\emptyset, \{x_1\}, \ldots, \{x_{n-1}\}, X$. □
Using this idea, a way of getting around the problem above is to append the symbol space, I, with a single new symbol, $\perp$. Then, we place the following topology on $I' = I \cup \{\perp\}$: $2^I, I \cup \{\perp\}$, where 2^I is the power set of I. This topology on I' makes it homeomorphic to X in the proof above (when X and I' have the same number of elements). Therefore, it is T_0, but not T_1. Now, we can create continuous maps from a continuum, Y, into I' as follows: Let A_i be N mutually disjoint open sets *not* covering Y. Let $I = \{1, \ldots, N\}$, and define $f(A_i) = i$ and $f(Y - \bigcup_i A_i) = \perp$. We claim f is continuous. It is enough to check the basis elements of the topology on I', which are the singleton sets of elements of I plus the set I' itself. We have $f^{-1}(i) = A_i$, open, for each $i \in I$. Further, $f^{-1}(I') = Y$, which is open.

Another topology which works is the following: $\emptyset, \{V \cup \{\perp\} \mid V \in 2^I\}$, with the A_i *closed* instead of open (see §6.3 for a use of a topology like this). There are presumably many other choices one can make. Below we examine at length one espoused in [114].

§6.2.2 *AD* MAP OF NERODE-KOHN

DEFINITION:

The **AD map** is a map from the measurement space, Y, into a finite set of symbols, I. Nerode and Kohn [114] create a continuous AD map as follows:

Algorithm 6.2 (Open Cover Topology) 1. First, take any finite open cover of the measurement space: $Y = \bigcup_{i=1}^n A_i$, where the A_i are open in the given topology of Y.

2. Next, find the so-called *small topology*, $\mathcal{T}_Y$, generated by the subbasis A_i. This topology is finite (as we argue below) and its open sets can be enumerated, say, as $B_1, \ldots, B_p$.

3. Next, find all the non-empty **join irreducibles** in the collection of the B_i (that is, all non-empty sets B_j such that if $B_j = B_k \cup B_l$, then either $B_j = B_k$ or $B_j = B_l$). Again, there are a finite number of such join irreducibles, which we denote $C_1, \ldots, C_N$.

4. Let the set of symbols be $I = \{1, \ldots, N\}$. Further, define the function $AD(y) = i$ if C_i is the smallest open set containing y.

5. Create a topology, $\mathcal{T}_I$, on I as follows. For each $i \in I$, declare $D_i = \{j \mid C_j \subset C_i\}$ to be open. Let $\mathcal{T}_I$ be the topology generated by the D_i.

Here is a simple example of the construction:

Example 6.3 *Let our measurement space be $Y = [0, 3]$ and the open cover of this measurement space be*

$$A_1 = [0, 2), \qquad A_2 = (1, 3].$$

The small topology generated by this subbasis can be enumerated as follows: $B_1 = \emptyset$, $B_2 = (1, 2)$, $B_3 = [0, 2)$, $B_4 = (1, 3]$, $B_5 = [0, 3]$. Next, we find the non-empty join irreducibles:

$$C_1 = [0, 2), \quad C_2 = (1, 3], \qquad C_3 = (1, 2).$$

Thus, we let our set of symbols be $I = \{1, 2, 3\}$ and define the function AD as follows:

$$AD(y) = \begin{cases} 1, & y \in [0, 1], \\ 2, & y \in [2, 3], \\ 3, & y \in (1, 2). \end{cases}$$

The open sets D_i are found to be

$$D_1 = \{1, 3\}, \qquad D_2 = \{2, 3\}, \qquad D_3 = \{3\}$$

and the resulting topology on I, $\mathcal{T}_I$ is

$$\emptyset, \qquad \{3\}, \qquad \{1, 3\}, \qquad \{2, 3\}, \qquad \{1, 2, 3\}.$$

One can readily check that $\mathcal{T}_I$ is T_0 and that AD is continuous. □

FILTER INTERPRETATION:

Here, we give an intuitive interpretation of the Nerode-Kohn approach to hybrid systems as described in [114] (herein, *N-K*) in terms of bandpass filters. Our discussion covers both *AD* and *DA* maps.

The starting point of the *N-K* approach is an assumption that one can only realistically distinguish points up to knowing the open sets in which they are contained. Thus, one takes small topologies on the measurement (a.k.a. plant output) and control (a.k.a. plant input) spaces. The open sets in these topologies correspond to events that are distinguishable and achievable, respectively. For example, they represent measurement error or actuator error (or equivalence classes that are adequate for the task at hand).

> NOTE. However, the theory developed from this principle is destined to contradict itself. In particular, we have seen that closed sets may be distinguished (these arise from the partition of the measurement space into symbol pre-images, the so-called "essential parts.") More provocatively, we can distinguish single points in the measurement space. Consider as a representative example zero in $[-1, 1]$. Then the open sets [-1,1], (0,1], and [-1,0) give us information to exactly deduce $x = 0$.

A good way to think of the open sets in the small topology is as notch filters. On the input side, we can pass our measurements through these filters. The level of information that we glean is, Did it go through the filter or not? Now, the total information from our sensors is summarized in the string of Yes/No answers.[1] (Of course, we also implicitly have the filters themselves, that link these binary symbols with real regions of measurement space.) By taking the intersection of all filters which had a Yes answer, we obtain the join irreducible from which the measurement came. The input symbols of the finite automaton are simply "names" given to join irreducibles. By also taking into account the No answers, we obtain a partition of the measurement space into what *N-K* call **essential parts**. See Figure 6-2

Likewise, on the output side one constructs the join irreducibles. The output symbols of the finite automata are exactly "names" given to these join irreducibles. Now, the finite

---

[1]That is, we can basically do peak detection (now allowing nonideal filters). Biological auditory and olfactory systems may work like this [77].
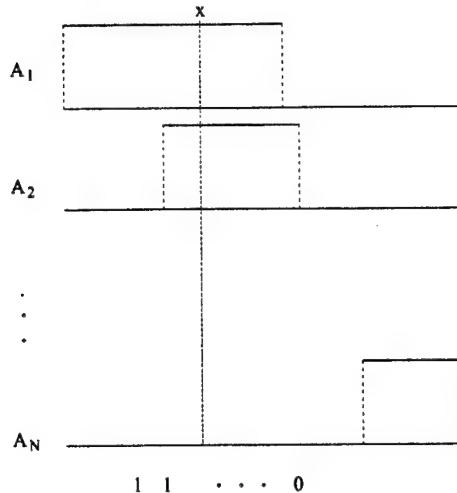
Figure 6-2: Filter interpretation of the open cover A_i.

automata controller is simply a map from input symbols to output symbols (modulated by its internal state). To fix ideas, let's say that the output symbol corresponds to join irreducible K_j.

Again, we can think of the control space small topology as a set of notch filters. Here, we imagine some broadband source signal (which is not exactly flat) which we use to produce our control in the following way: Instead of choosing a single output from the named join irreducible deliberately (normal AD conversion), we simply construct one in the correct equivalence class. We do this by using as a control signal the signal that results from passing our broadband source through each of the filters (open sets) which intersect to form the join irreducible K_j.

It is also interesting to note that *N-K* seem to have adopted the idea (cf. Appendix II of [114]) that the finite automaton and small topologies are used to construct approximations to maps from the measurement to control spaces, the approximation (of a continuous control law) necessarily approaching that law as the cover becomes finer.

VERIFICATION OF CONTINUITY:

One of the results of [114] is the fact that their AD map is continuous from $(Y, \mathcal{T}_Y)$ to $(I, \mathcal{T}_I)$. Namely, they give (without proof) the following proposition, whose proof *we* provide for completeness:

Proposition 6.4 (Nerode-Kohn) $AD : Y \mapsto I$ *is continuous.*

We need several technical lemmas first, which are also used to prove later results:

Lemma 6.5 *The non-empty join irreducibles of the topology generated by the (subbasis) A_i are exactly those sets which can be written as*

$$V_y = \bigcap_{j \in J(y)} A_j,$$

where $J(y)$ is the set of all j such that $y \in A_j$.

Proof.   Pick $y \in Y$ arbitrarily. Then V_y is non-empty since it contains y. Next, suppose that V_y is not join irreducible, so that it can be written as $V_y = A \cup B$ where $A \neq V_y$ and $B \neq V_y$. Then we must have that $y \in A$ or $y \in B$, or both. Without loss of generality, assume $y \in A$. But, since A is an element of the topology generated by the A_i, it must be of the form $A = \bigcup_{i \in I} \bigcap_{j \in J_i} A_j$, where $J_i \subset \{1, \ldots, n\}$, for each $i \in I$, that is, an arbitrary union of finite intersections of elements of the subbasis. Since $y \in A$, it must be in at least one of the sets in the union, say, the kth: $y \in \bigcap_{j \in J_k} A_j$. However, this means $y \in A_j$ for each $j \in J_k$. By definition, $J_k \subset J(y)$, so that $V_y \subset A$. But, since $V_y = A \cup B$, we also have $V_y \supset A$. So that $A = V_y$, a contradiction. $\square$

Thus, AD is a well-defined function, with $AD(y) = i$ where C_i is the smallest join irreducible containing y. In fact, C_i equals the V_y defined in the lemma. Now, it is easy to see that the set of non-empty join irreducibles is finite: there are at most $2^n - 1$ distinct non-empty sets that can be written in this manner. Thus, the topology generated by the A_i has less than 2^{2^n} elements (since each element is a union of basis sets).

Lemma 6.6   *1. The non-empty join irreducibles C_i form a basis of the topology, $\mathcal{T}_Y$, which they generate.*

2. The sets D_i are a basis for the topology, $\mathcal{T}_I$, which they generate.

3. AD is surjective.

4. If f is surjective, $f(f^{-1}(X)) = X$.

5. $C_j = AD^{-1}(D_j)$.

Proof.   The detailed proofs appear in the Appendix. Items 1 through 3 are straightforward. (For those with a knowledge of lattice theory, the C_i and D_i are lower closures in their respective lattices and give rise to the (dual) Alexandrov topologies thereupon [141].) Item 4 is in [113, p. 20]. Item 5 is almost immediate in the $\supset$-direction and follows with the help of Lemma 6.5 in the $\subset$-direction. $\square$

Now, we are ready to prove the proposition:

Proof.   (of Prop. 6.4) Lemma 6.6 says the D_j are a basis and that $AD^{-1}(D_j) = C_j$, which is open in Y. $\square$

$\mathcal{T}_I$ IS THE QUOTIENT TOPOLOGY:

Next, we want to show that the AD topology of Nerode and Kohn, $\mathcal{T}_I$, is exactly the quotient topology of their AD map. This is accomplished by proving that $\mathcal{T}_I$ is both coarser and finer than the quotient topology. The following is well-known [113, p. 143]:

> Let X be a space; let A be a set; let $p : X \mapsto A$ be a surjective map. Then the quotient topology on A induced by p is the finest (i.e., largest) topology relative to which p is continuous.

Since the AD map is continuous in $\mathcal{T}_I$ and surjective, we trivially have: $\mathcal{T}_I$ *is coarser than the quotient topology*, $\mathcal{T}_Q$, *corresponding to AD.* Now, it remains to show that $\mathcal{T}_I$ *is finer than $\mathcal{T}_Q$.* **Proof.**   Suppose J is open in $\mathcal{T}_Q$. Then $AD^{-1}(J)$ is open in $\mathcal{T}_Y$. Finally, it can be written as $AD^{-1}(J) = \bigcup_{\beta \in B} C_\beta$, where B is some subset of $\{1, \ldots, N\}$, since the C_β are

a basis for $\mathcal{T}_Y$. We want to show that $J \in \mathcal{T}_I$. But note that since AD is surjective

$$J = AD(AD^{-1}(J)) = AD\left(\bigcup_{\beta \in B} C_\beta\right) = \bigcup_{\beta \in B} AD(C_\beta).$$

Now, we have from Lemma 6.6 that $C_\beta = AD^{-1}(D_\beta)$. Since AD is surjective, this implies $AD(C_\beta) = AD(AD^{-1}(D_\beta)) = D_\beta$ So that $J = \bigcup_{\beta \in B} D_\beta$, which is open in $\mathcal{T}_I$, being a union of basis elements. $\square$

Summarizing, we have shown

Theorem 6.7 *The AD topology of Nerode and Kohn, $\mathcal{T}_I$, is exactly the quotient topology of their AD map.*

We have also gotten something else along the way. In the last proof we showed that $AD(C_\beta) = D_\beta$. From Lemma 6.6, we have $AD^{-1}(D_\beta) = C_\beta$ and that D_β and C_β are bases for $\mathcal{T}_I$ and $\mathcal{T}_Y$, resp. Thus, AD is a homeomorphism between the topological spaces $(Y, \mathcal{T}_Y)$ and $(I, \mathcal{T}_I)$. (This homeomorphism was also noted without proof in [114].)

§6.3   COMPLETING THE LOOP

§6.3.1   PROBLEMS COMPLETING THE LOOP

In this section, we discuss problems which arise when considering continuous mappings from the measurement to control spaces (see Figure 6-1). Specifically, we have

Remark 6.8 *If Y is connected and U is T_1, the only continuous maps from Y to a finite subset of U (i.e., $f(Y) = \{u_1, \ldots, u_N\}$, $u_1, \ldots, u_N \in U$) are constant maps.*

Proof.   First, constant maps are always continuous, and their image is a single point of U, hence finite. Next, suppose for contradiction that f is a non-constant continuous map from Y into U and the image $f(Y) = \{u_1, \ldots, u_N\}$, where the u_i are distinct points in U for some finite N greater than or equal to two. Since U is T_1, we can construct open sets $V_{i,j}$ for $i \neq j$ such that $V_{i,j}$ contains u_i but not u_j. Thus, there is an open set about u_1 not containing $u_2, \ldots, u_N$, viz., $V = \bigcap_{j=2}^N V_{1,j}$. Also, we can construct an open set which contains each $u_2, \ldots, u_N$ yet does not contain u_1: $W = \bigcup_{i=2}^N V_{i,1}$. Therefore, $f(Y) = V \cup W$ is not connected. $\square$

§6.3.2   TOPOLOGIES COMPLETING THE LOOP

In the previous subsection, we saw that, under mild assumptions, there are no non-constant continuous maps from the measurement to control spaces. In this subsection, we wish to give a topology on the (augmented) control space which allows us to construct a non-constant continuous map.

We make no assumptions on Y and U (except those implicit in the definition of f below). Suppose that the topology on U is $\mathcal{T}$. Then we let $U' = U \cup \{\perp\}$, that is, we append a single element, $\perp$, to U. Next, we define a topology, $\mathcal{T}'$, on U' as follows: $\mathcal{T}' = \emptyset, \{V \cup \{\perp\} \mid V \in \mathcal{T}\}$. Suppose we wish to have image points $u_1, \ldots, u_N$ in U. Let $f^{-1}(u_i) = K_i$ be disjoint closed sets not covering Y. Let $f(Y - \bigcup_{i=1}^N K_i) = \perp$. Then

Remark 6.9 *f is continuous.*

Proof. $f^{-1}(\emptyset) = \emptyset$, which is open. Now, suppose V' is any non-empty open set of U'. Then $V' = V \cup \{\bot\}$, where V is open in U. Therefore,

$$
\begin{aligned}
f^{-1}(V') &= f^{-1}(V) \cup f^{-1}(\{\bot\}) \\
&= \left(\bigcup_{j \in J} K_j \right) \cup \left(Y - \bigcup_{i \in I} K_i \right) \\
&= Y - \bigcup_{i \in I-J} K_i,
\end{aligned}
$$

(where J is the set of indices j for which $u_j \in V$, and $I = \{1, \dots, N\}$) which is open since its complement is closed; the formula is well-defined if $J \neq I$. If $J = I$, then $f^{-1}(V') = Y$, which is open. $\qquad\square$

§6.4   A DIFFERENT VIEW OF HYBRID SYSTEMS

We wish to propose a different view of hybrid systems as shown in Figure 6-3.
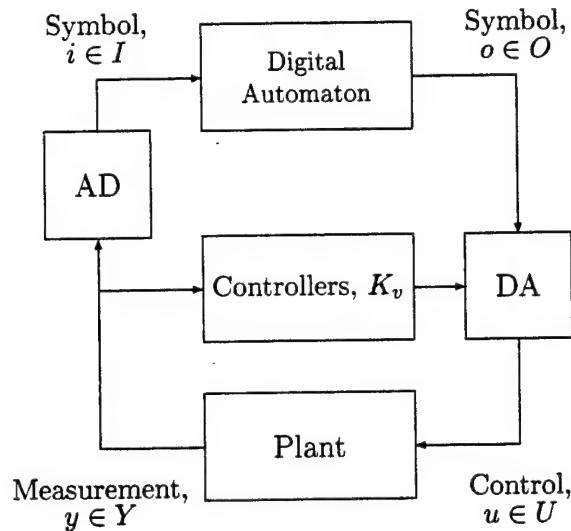
Figure 6-3: Alternative prototypical hybrid system.

The difference between this and the previous prototypical hybrid system is that there is feedback on the signal level. This feedback modulates the symbols coming down from the higher level. Alternatively, one can view the symbols as specifying one of several controllers whose output is to be the control signal.

The most widely used fuzzy control scheme is related to this model in the sense that it achieves continuous maps—despite a finite number of so-called fuzzy rules—by utilizing the continuous measurement information. We discuss this in more detail below.

§6.4.1   WHY THE DIFFERENT VIEW?

Before, we had a natural fan-in of sensory information from the signal to symbol levels. This models abstraction and reduction. In our new view, we also have an analogous, natural fan-out of control commands from the symbol to signal level that was not present before.

Basically, we are saying that the finite description of the plant's dynamics as seen from automaton's point of view is not an exact aggregation of the plant's dynamics. Therefore, one should utilize the continuous information present at the lower level as well as the discrete decision made above in order to choose a control input for the lower level. There is no need to arbitrarily pick a member from the set of controls (fixed for normal AD conversion, always arbitrary in the Nerode-Kohn view). Instead, the *set* is given by the automaton, while the *member* of that set is chosen using information from the lower level. Thus, the aggregated and continuous dynamics are related, but the first is not a substitute for the latter. If it were, the plant could have been modeled directly as a finite automaton.

§6.5 EXAMPLE OF THE DIFFERENT VIEW: FUZZY CONTROL

We now wish to examine the different view of hybrid systems as shown in Figure 6-3. Such a view can give rise to a continuous map completing the loop. In particular, the most widely used fuzzy control scheme is related to this model in the sense that it achieves continuous maps—despite a finite number of so-called fuzzy rules—by utilizing the continuous measurement information. We discuss this in more detail below.

THE CONTROL SCHEME:

A fuzzy control scheme is given by the commuting diagram of Figure 6-4, where F denotes fuzzification, G the inference map of the fuzzy rule base, and D defuzzification. Here, the

$$
\begin{array}{ccc}
x \in X & \xrightarrow{\;\;g\;\;} & z \in Z \\[4pt]
\Big\downarrow{\scriptstyle F} & & \Big\uparrow{\scriptstyle D} \\[4pt]
\begin{array}{c}[\mu_{A_1}(x),\cdots,\mu_{A_M}(x)]\\ \in [0,\,1]^M\end{array} & \xrightarrow{\;\;G\;\;} & \begin{array}{c}[\mu_{B_1'}(x),\cdots,\mu_{B_M'}(x)]\\ \in \mathcal{F}^M\end{array}
\end{array}
$$
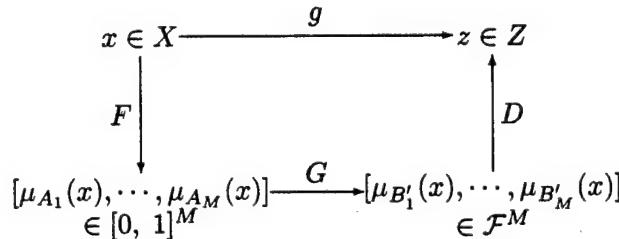
Figure 6-4: Fuzzy Logic Controller.

fuzzy controller has M rules of the form

$$\text{RULE}_i : \text{ IF } x \text{ is } A_i, \text{ THEN } z \text{ is } B_i, \quad i \in \{1,\ldots,M\}$$

and $\mathcal{F}^M$ is a cross product of the space of fuzzy sets on Z. The most widely used inference rule computes $\mu_{B_i'}(z) = \min\{\mu_{A_i}(x),\ \mu_{B_i}(z)\}$, for all $z \in Z$, and defuzzifies using the centroid: $z = (\sum_{i=1}^{M} \overline{z}^i\, \mu_{B_i'}(\overline{z}^i))/(\sum_{i=1}^{M} \mu_{B_i'}(\overline{z}^i))$, where $\overline{z}^i$ equals the centroid of $\mu_{B_i'}$: $\overline{z}^i = (\int_Z z\, \mu_{B_i'}(z)dz)/(\int_Z \mu_{B_i'}(z)dz)$.

The finite rule base is related to the finite symbols of our hybrid model. For instance, the rules which fire are akin to the filters which passed data in our discussion of the Nerode-Kohn approach. However, here one utilizes the underlying continuous information (represented in the continuous membership functions for the measurement space) in order to construct the precise control output. Thus, it fits into our different view. See Figure 6-5
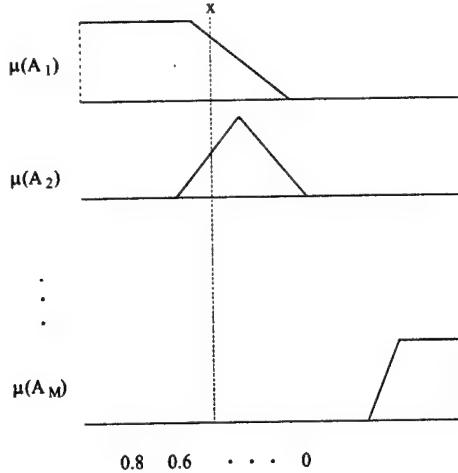
Figure 6-5: Filter interpretation of the fuzzy rule antecedents A_i.

PRODUCING CONTINUOUS MAPS:

We deal with the prototypical case where X and Z are closed intervals in $\mathbb{R}$ (for specificity, $[a, b]$ and $[c, d]$, resp.). The case where X is a multi-interval in $\mathbb{R}^m$ is a straightforward extension. The case where z is a multi-interval in $\mathbb{R}^n$ then follows from considering each dimension componentwise. We claim that the induced map $g = D \circ G \circ F$ is continuous from X to Z. We assume, for the proof, that the μ_{A_i} and μ_{B_i} are continuous on X and Z, resp. This is fairly typical (e.g., triangular functions).

Proof. If the μ_{A_i} are continuous, then F is continuous. It is also easy to see that centroid defuzzification, D, is continuous. It remains to show that G is continuous. Well, a fuzzy inference rule gives rise to the following situation: $H_{\alpha, f}(z) = \min\{\alpha, f(z)\}$, where $H_{\alpha, f}$, f, and α are playing the role of fixed $\mu_{B'_i}$, μ_{B_i}, and $\mu_{A_i}(x)$ resp. Thus, by assumption, $f(z) \in C([c, d] \to [0, 1])$. Now, we need G to be continuous as a map from, componentwise, $[0, 1]$ to $C([c, d] \to [0, 1])$. But, if $|\alpha_1 - \alpha_2| < \epsilon$, then $\|H_{\alpha_1, f} - H_{\alpha_2, f}\| < \epsilon$ where $\|\cdot\|$ denotes the sup norm. $\qquad\square$

APPROXIMATING CONTINUOUS MAPS:

Fuzzy control maps are also dense in the set of continuous functions from X to Z. It is enough to note that triangular functions, which are prevalent for descriptions of fuzzy membership sets, are so dense. To more easily see this, note that triangular functions can be combined to construct arbitrary piecewise linear functions.

§6.6 A UNIFYING TOPOLOGICAL VIEWPOINT

As open and closed sets play dual roles in topology, we can define topologies based on *closed* sets by replacing "open" with "closed" in Algorithm 6.2. Of course, one still runs into the problem that essential parts may be points. However, we provide a construction below that avoids this. Further, it relates the topological results of this chapter back to our assumptions for the existence and uniqueness of hybrid systems trajectories and to insuring finitely many jumps in finite time [§5]. Further, it is related to our simulation definitions in §7.

For sake of argument, assume that the measurement space is a bounded metric space. Then the following two constructions are dual.

- Pick as open cover, Y plus a finite number of disjoint open sets separated by at least some distance ϵ.

- Pick as "closed cover," Y plus the disjoint closures of a finite number of open sets.

In each case, $Y - \bigcup_i A_i$ is mapped to "continue" and the others to different symbols. The advantages of the latter construction are that: (1) each AD_q map is continuous; (2) it is consistent with the conditions for the dynamics to be uniquely defined.

§6.7 NOTES

The majority of this chapter appeared in [24]. The topological view of §6.6 did not appear but was presented in the associated conference talk.

The filter interpretation of the Nerode-Kohn approach arose from discussions with Anil Nerode.

§6.8 APPENDIX

This appendix collects the full proofs for statements 1–3 and 5 in Lemma 6.6. They are listed as separate lemmas for convenience.

Lemma 5.1 *The non-empty join irreducibles C_i form a basis of the topology, $\mathcal{T}_Y$, which they generate.*

Proof. Each $y \in Y$ is contained in such a set since the A_i are a cover of Y. The intersection of two such sets that contain the point y is a superset of V_y. □

Lemma 5.2 *The sets D_i are a basis for the topology, $\mathcal{T}_I$, which they generate.*

Proof. Each $i \in I$ is contained in D_i since $C_i \subset C_i$, so there is a basis element containing each $i \in I$. If $i \in I$ belongs to the intersection of two basis elements, say D_{j_1} and D_{j_2}, then we need a basis element D_{j_3} containing i such that $D_{j_3} \subset D_{j_1} \cap D_{j_2}$. But then $C_i \subset C_{j_1}$ and $C_i \subset C_{j_2}$. So that $C_i \subset C_{j_1} \cap C_{j_2}$. From this, we want to show that D_i is contained in $D_{j_1} \cap D_{j_2}$. But this is evident from the definition of D_i:

$$D_i = \{j \mid C_j \subset C_i\}.$$

So, if $j \in D_i$, then $C_j \subset C_i \subset C_{j_1}$, so that $j \in D_{j_1}$. Likewise, $C_j \subset C_i \subset C_{j_2}$, so that $j \in D_{j_2}$. Therefore, $D_i \subset D_{j_1} \cap D_{j_2}$, is the required basis element. □

Lemma 5.3 *AD is surjective.*

Proof. Pick $i \in I$. By construction, there exists $y \in Y$ such that C_i is the smallest non-empty join irreducible containing y. $AD(y) = i$. □

Lemma 5.5 $C_j = AD^{-1}(D_j)$

Proof.

1. $C_j \supset AD^{-1}(D_j)$.

$$
\begin{aligned}
AD^{-1}(D_j) &= AD^{-1}(\cup_{k \in D_j} k) \\
&= \bigcup_{k \in D_j} AD^{-1}(k) \subset \bigcup_{k \in D_j} C_k \subset C_j.
\end{aligned}
$$

The last inequality follows from the fact that $C_k \subset C_j$ for all $k \in D_j$.

2. $C_j \subset AD^{-1}(D_j)$. Suppose $y \in C_j$. Then either C_j is the smallest non-empty join irreducible containing y, in which case we are done, or there is some other smallest non-empty join irreducible C_k containing y. We claim $C_k \subset C_j$, in which case $k \in D_j$ and $y \in AD^{-1}(D_j)$, which is the desired result.

Therefore, it remains to show that $C_k \subset C_j$. The smallest join irreducible containing y is (see Lemma 6.5) is $V_y = \bigcap_{j \in J(y)} A_j$ where $J(y)$ is the set of all j such that $y \in A_j$. However, C_j is also a join irreducible, so that it can be written $C_j = \bigcap_{j \in J} A_j$ for some $J \subset \{1, \ldots, n\}$. But C_j contains y, so that each of the A_j in the intersection must contain y. So that by definition $J \subset J(y)$, whence $C_k \equiv V_y \subset C_j$.

$\square$

Chapter 7

Complexity and Computation in Hybrid Systems

We explore the simulation and computational capabilities of hybrid and continuous dynamical systems. Notions of simulation of a discrete transition system by a continuous one are developed. We show that hybrid systems whose equations allow a precise binary timing pulse (exact clock) can simulate arbitrary reversible discrete dynamical systems defined on closed subsets of $\mathbb{R}^n$. We also prove that any discrete dynamical system in $\mathbb{Z}^n$ can be simulated by continuous ODEs in $\mathbb{R}^{2n+1}$. We use this to show that there are smooth ODEs in $\mathbb{R}^3$ that possess the power of universal computation. We use the famous asynchronous arbiter problem to distinguish between hybrid and continuous dynamical systems.

§7.1 INTRODUCTION

In this chapter, we explore the simulation and computational capabilities of hybrid systems. This chapter is a step towards the characterization of these models in terms of the types of systems that can be described by, or "implemented" with, their equations. By construction, however, a hybrid system model can implement ODEs with continuous vector fields (continuous ODEs). Thus, even with no discrete dynamics, these models can describe a large variety of phenomena.

In addition to "implementing" ODEs, all reviewed models can implement a precise binary timing pulse or "exact clock" (defined later). Thus, we explore the capabilities of systems with continuous ODEs and exact clocks. For instance, we show such systems can simulate arbitrary reversible discrete dynamical systems defined on closed subsets of $\mathbb{R}^n$. These simulations require ODEs in $\mathbb{R}^{2n}$ which use an exact clock as input.

Later, we find that one can still simulate arbitrary discrete dynamical systems defined on subsets of $\mathbb{Z}^n$ without the capability of implementing an exact clock: one can use an approximation to an exact clock. Such an "inexact clock" is implemented with continuous functions of the state of a one-dimensional continuous ODE. As a result, one can perform such simulations using continuous ODEs in $\mathbb{R}^{2n+1}$. Turning to computational abilities, we show that continuous ODEs in $\mathbb{R}^3$ possess the ability to simulate arbitrary Turing machines, pushdown automata, and finite automata. By simulating a universal Turing machine, we conclude that there exist ODEs in $\mathbb{R}^3$ with continuous vector fields possessing the power of universal computation. Further, the ODEs simulating these machines may be taken smooth and do not require the machines to be reversible (cf. [108]).

Finally, we show that hybrid dynamical systems are strictly more powerful than Lipschitz ODEs in the types of systems they can implement. For this, we use a nontrivial example: the

famous asynchronous arbiter problem [26, 100, 144]. First we quickly review the problem. Then we settle it in an ODE framework by showing that one cannot build an arbiter out of devices modeled by Lipschitz ODEs. Next, we examine the problem in a hybrid systems framework. We show that all the hybrid systems of §3 can implement an arbiter even if their continuous dynamics is a system of Lipschitz ODEs.

The chapter is organized as follows. In §7.2 notions of simulation are discussed. Here, we make precise what we mean by "simulation" of discrete transition systems by continuous transition systems. All our simulation results are collected in §7.3. §7.4 deals with the asynchronous arbiter problem. The Appendix collects some technical lemmas.

§7.2 NOTIONS OF SIMULATION

In dynamical systems, simulation is captured by the notions of topological equivalence and homomorphism [59, 67, 129]. One can extend these notions to systems with inputs and outputs by also allowing memoryless, continuous encoding of inputs, outputs, and initial conditions.

In computer science, simulation is based on the notion of "machines that perform the same computation." This can be made more precise, but is not reviewed here [20, 104].

Other notions of simulation (for discrete dynamical systems) appear in [91]. All these notions, however, are "homogeneous," comparing continuous systems with continuous ones or discrete with discrete. One that encompasses simulation of a discrete transition system by a continuous transition system is required here.

One notion that associates discrete and continuous transition systems is global section [129]. The set $S_X \subset X$ is a **global section** of the continuous dynamical system $[X, \mathbb{R}_+, f]$ if there exists a $t_0 \in \mathbb{R}_+$ such that

$$S_X = \{f(P, kt_0) \mid k \in \mathbb{Z}_+\},$$

where P is a set containing precisely one point from each of the trajectories $f(p, \mathbb{R}_+)$, $p \in X$. Using this for guidance, we define

Definition 7.1 (S-simulation) *A continuous transition system $[X, \mathbb{R}_+, f]$* **simulates via section** *or* **S-simulates** *a discrete transition system $[Y, \mathbb{Z}_+, F]$ if there exist a continuous surjective partial function $\psi : X \to Y$ and $t_0 \in \mathbb{R}_+$ such that for all $x \in \psi^{-1}(Y)$ and all $k \in \mathbb{Z}_+$*

$$\psi(f(x, kt_0)) = F(\psi(x), k).$$

Note that surjectivity implies that for each $y \in Y$ there exists $x \in \psi^{-1}(y)$ such that the equation holds. Here, continuous partial function means the map from $\psi^{-1}(Y)$ (as a subspace of X) to Y is continuous.

Intuitively, the set $V \equiv \psi^{-1}(Y)$ may be thought of as the set of "valid" states; the set $X \backslash V$ as the "don't care" states. In dynamical systems, V may be a Poincaré section; $X \backslash V$ the set of points for which the corresponding Poincaré map is not defined [67, 73]. In computer science and electrical engineering, V may be the set of circuit voltages corresponding to a logical 0 or 1; $X \backslash V$ the voltages for which the logical output is not defined.

S-simulation is a strong notion of simulation. For instance, compare it with topological equivalence. Typically, though, the homogeneous notions of simulation do not expect time to be parameterized the same (up to a constant) for both systems. For example, a universal Turing machine, U, may take several steps to simulate a single step of any given Turing

machine, M. Moreover, the number of such U steps to simulate an M step may change from M step to M step. Some of the notions of simulation defined in [91] also allow this generality. Further, the definition of topological equivalence of vector fields (different than for dynamical systems, see [67]) is such that parameterization of time need not be preserved. Thus, following the definitions in [91] one formulates

Definition 7.2 (P-simulation) *A continuous transition system* $[X, \mathbb{R}_+, f]$ **simulates via points** *or* **P-simulates** *a discrete transition system* $[Y, \mathbb{Z}_+, F]$ *if there exists a continuous surjective partial function* $\psi : X \to Y$ *such that for all* $x \in \psi^{-1}(Y)$ *there is a sequence of times* $0 = t_0 < t_1 < t_2 < \cdots$, $\lim_{k \to \infty} t_k = \infty$, *such that*

$$\psi(f(x, t_k)) = F(\psi(x), k).$$

One readily checks that S-simulation implies P-simulation. This is a weak notion. For instance, consider the case where Y is finite, $|Y| = N$. Suppose $[X, \mathbb{R}_+, f]$ has a point p such that $|f(p, \mathbb{R}_+)| \geq N$ and $p = f(p, t_0)$ for some $t_0 > 0$. That is, the orbit at point p is periodic and contains more than N points. Clearly, one may associate N distinct points in $f(p, \mathbb{R}_+)$ with the points in Y, so that $[X, \mathbb{R}_+, f]$ P-simulates $[Y, \mathbb{Z}_+, F]$. This weakness persists even if Y is infinite. For example, the simple harmonic oscillator defined on the unit circle, $X = S^1$:

$$
\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= -x_1,
\end{aligned}
$$

along with $\psi(x) = x_1$ P-simulates every $[[-1, 1], \mathbb{Z}_+, F]$. These arguments also show the weakness of some of the definitions in [91]. Finally, this same example shows P-simulation does not imply S-simulation: the harmonic oscillator above cannot S-simulate any $[[-1, 1], \mathbb{Z}_+, F]$ for which 0 is a fixed point and 1 is not a fixed point.

Thus, P-simulation need not correspond to an intuitive notion of simulation. The reason is that one wants, roughly, homeomorphisms from orbits to orbits, not from points to points. As mentioned in §2.1, this is achieved with continuous dynamical systems. However, this is not possible with nontrivial nonhomogeneous systems since a discrete orbit with more than one point is a (countable) disconnected set and any non-constant continuous orbit is an (uncountable) connected set. Thus, there exist homeomorphisms between discrete and continuous orbits only when both are constant.

If X is connected and Y is a discrete topological space, this situation exists even with points, i.e., the only continuous functions from X to Y are constant functions [113]. One way to remedy this is simply to place topologies on X and Y other than their usual topologies, so that continuous maps are possible (cf. §3.6). There are several ways to accomplish this. One approach is to use so-called small topologies on X. Another is to append a single element $\{\perp\}$ to Y, which stands for "don't care" or "continue," and topologize $Y' = Y \cup \{\perp\}$. For more information and other approaches see [24, 114, §6].

Here—and with a view towards simulating systems defined on discrete topological spaces—we strengthen the definition of P-simulation in two ways. First, we require that the "simulated state" be valid on some neighborhood and for at least some minimal time period. Physically, this allows one to use "imprecise sampling" to obtain discrete data, providing a robustness that is lacking in the definition of P-simulation. Second, we require that the "readout times" are exactly those for which $x(t) \in \psi^{-1}(Y)$.

Definition 7.3 (I-simulation) *A continuous transition system* $[X, \mathbb{R}_+, f]$ *simulates via intervals or* **I-simulates** *a discrete transition system* $[Y, \mathbb{Z}_+, F]$ *if there exist a continuous surjective partial function* $\psi : X \to Y$ *and* $\epsilon > 0$ *such that* $V \equiv \psi^{-1}(Y)$ *is open and for all* $x \in V$ *the set* $T = \{t \in \mathbb{R}_+ \mid f(x, t) \in V\}$ *is a union of intervals* (τ_k, τ_k'), $0 = \tau_0 < \tau_0' < \tau_1 < \tau_1' < \cdots$, $|\tau_k' - \tau_k| \geq \epsilon$, *with*

$$\psi(f(x, t_k)) = F(\psi(x), k),$$

for all $t_k \in (\tau_k, \tau_k')$.

Clearly I-simulation implies P-simulation. S-simulation and I-simulation, however, are independent notions.

The extra requirement that $\psi^{-1}(Y)$ be open implies that the inverse images of open sets in Y are open in X (and not just in $\psi^{-1}(Y)$ as before). This is probably too strong a requirement in the case of a general topological space Y. However, in the case of Y a discrete topological space, it has the desirable effect that $\psi^{-1}(y)$ is open for all $y \in Y$.

One might also have required an output map that is zero (or any distinguished output value) on the complement of T and non-zero otherwise. This amounts to, in the case of a universal Turing machine simulating a machine M, the existence of a distinguished state meaning "a step of the simulated machine is not yet completed." Here, it is related to the appending of a symbol $\{\perp\}$ to Y as above and extending $\psi : X \to Y' = Y \cup \{\perp\}$ by defining $\psi(x) = \{\perp\}$ if $x \in X \backslash \psi^{-1}(Y)$ [7, 24, 114]. In this case, the requirements on ψ may be replaced by requiring ψ to be continuous from X to Y' (in a suitable topology) after extension. Finally, if X is a metric space one could introduce a "robust" version of I-simulation by requiring the inverse image of $y \in Y$ to contain a ball with at least some minimum diameter.

Below, "simulation" is a generic term, meaning I-simulation, S-simulation, or both. SI-simulation denotes S-simulation and I-simulation. If a machine is equivalent, or simulates one that is equivalent, to a universal Turing machine, one says it has the **power of universal computation.**

§7.3 SIMULATION WITH HYBRID & CONTINUOUS SYSTEMS

In this section we concentrate on general simulation results and the capabilities of hybrid systems and continuous ODEs.

We first construct low-dimensional discrete dynamical systems in $\mathbb{Z}^n$ that are equivalent to finite automata (FA), pushdown automata (PDA), and Turing machines (TMs). Later, we give some general results for continuous ODEs in $\mathbb{R}^{2n+1}$ simulating discrete dynamical systems in $\mathbb{Z}^n$. Combining allows us to conclude simulation of arbitrary FA, PDA, and TMs. By simulating a universal TM, one obtains continuous ODEs with the power of universal computation. In the process, we also discuss the simulation and computational capabilities of hybrid systems.

§7.3.1 DISCRETE DYNAMICAL SYSTEMS EQUIVALENT TO FA AND TMs

We start by showing that every TM is equivalent to a discrete dynamical system in $\mathbb{Z}^2$ and then consider systems equivalent to PDA and FA. Later, we refine these results to discrete dynamical systems in $\mathbb{Z}$ equivalent to TMs, PDA, and FA.

The FA, PDA, and TMs considered here are deterministic. Thus their transition functions naturally give rise to discrete dynamical systems. These are defined on state spaces of input strings and states; input strings, states, and stacks; and states, tape head positions, and tapes, respectively.

Here, the states, input strings, stacks, and tape configurations of automata and Turing machines are taken in the discrete topology; $\mathbb{Z}^n$ as a topological or normed space is considered as a subspace of $\mathbb{R}^n$ (in particular, it has the discrete topology).

See [76, §2.1.4] for precise definitions of FA, PDA, and TM.

Proposition 7.4   *1. Every TM is equivalent to a discrete dynamical system in $\mathbb{Z}^2$.*

2. *There is a discrete dynamical system in $\mathbb{Z}^2$ with the power of universal computation.*

3. *Every FA and inputless PDA is equivalent to a discrete dynamical system in $\mathbb{Z}$. Every PDA is equivalent to a discrete dynamical system in $\mathbb{Z}^2$.*

Proof.

1. Assume the tape alphabet is $\Gamma = \{\gamma_0, \gamma_1, \ldots, \gamma_{m-2}\}$, $m \geq 2$, with γ_0 the blank symbol; and that the set of states is $Q = \{q_0, \ldots, q_{n-1}\}$, $n \geq 1$. Define $p = \max\{m, n\}$.

   As is customary, the one-sided infinite tape is stored in two stacks, with the state stored on the top of the right stack. The coding used is p-ary. In particular, suppose the TM is in configuration C, with tape

   $$\mathcal{T} = \gamma_{i_1}, \ldots, \gamma_{i_{N-1}}, \gamma_{i_N}, \gamma_{i_{N+1}}, \ldots,$$

   head positioned at cell N, and internal state q_j. Encode the configuration C in the integers

   $$T_L = f_1(C) = \sum_{k=0}^{N-1} p^k i_{N-k} + p^N(m-1), \qquad T_R = f_2(C) = j + \sum_{k=1}^{\infty} p^k i_{N+k}.$$

   The second sum is finite since only finitely many tape cells are non-blank. The integer $(m-1)$ is an end-of-tape marker. The TM is assumed to halt on moving off the left of the tape, so that $(m-1, T_R)$ in $\mathbb{Z}^2$ is a fixed point for all valid T_R. On all other valid configurations, C, define transition function G in $\mathbb{Z}^2$ by

   $$G(f_1(C), f_2(C)) = (f_1(C'), f_2(C')),$$

   where C' is the configuration resulting when the next move function of the TM is applied to configuration C.

2. Use part 1 with any universal TM.

3. The inputless cases are immediate from part 1. For the cases with input, note that we encode the input string in an integer like the left part of the tape of a TM above, the results following.

   □

Note that one can perform the above encodings of TMs, FA, and PDA with $[0, p]$ replacing $\mathbb{Z}$. Merely replace p by p^{-1} in the formulas. The important thing added is compactness,

and other encodings, e.g., with $[0,1]$ replacing $\mathbb{Z}$, follow similarly. There is a problem using these encodings since two distinct tapes may have the same encoding, e.g., $3, 2, 0^\omega$ and $3, 1^\omega$. One can get around this by "separating" each tape encoding by replacing p with $2p$ and using $2i$ for the ith symbol. Namely, the tape of length N, $\mathcal{T} = \gamma_{i_1}, \ldots, \gamma_{i_N}$, is encoded as $\sum_{k=0}^{N}(2p)^{-k}2i_k$. Such **Cantor encodings** were used in [131]. We still do not use such encodings here, however, since later we want to ensure a minimum distance between any two tape encodings.

Finally, a wholly different approach is to use encodings inspired by those in [49]. Suppose we are given an arbitrary TM, T. Let q, h, l, and r be integer codings of its state, position of its read-write head, the parts of the tape on the left and on the right of its head, respectively. A configuration of T is encoded in the integer $2^q 3^h 5^l 7^r$.

More generally, any discrete dynamical system in $\mathbb{Z}^n$ is equivalent to one in $\mathbb{Z}$ by using such encodings, viz., by associating $(i_1, i_2, \ldots, i_n)$ with $p_1^{i_1} p_2^{i_2} \ldots p_n^{i_n}$, where p_i is the ith prime.

We could have used such constructions instead of those in Proposition 7.4. However, we retain them since their transition functions have properties which those arising from the "prime encodings" do not (cf. §7.3.3). In any case, we conclude

Proposition 7.5 *Every TM, PDA, inputless PDA, FA, and inputless FA is equivalent to a discrete dynamical system in $\mathbb{Z}$. There is a discrete dynamical system in $\mathbb{Z}$ with the power of universal computation.*

It is important to note that one can extend the transition functions in $\mathbb{Z}^n$ above to functions taking $\mathbb{R}^n$ to $\mathbb{R}^n$. We may extend any function $f : A \subset \mathbb{Z}^n \to \mathbb{R}^m$ in such a manner, by first extending arbitrarily to domain $\mathbb{Z}^n$ and then using linear interpolation. Here is an example, used below:

Example 7.6 *A continuous* mod *function may be defined as follows:*

$$x \bmod_C m \equiv \begin{cases} (\lfloor x \rfloor \bmod m) + x - \lfloor x \rfloor, & 0 \le \lfloor x \rfloor \bmod m < m-1, \\ (m-1)(\lfloor x \rfloor + 1 - x), & \lfloor x \rfloor \bmod m = m-1. \end{cases}$$

Later results require extensions that are robust to small input errors. That is, one would like to obtain the integer-valued result on a neighborhood of each integer in the domain. For instance, one may define a continuous nearest integer function, $[\cdot]_C : \mathbb{R} \to \mathbb{R}$, that is robust in this manner as follows:

$$[x]_C \equiv \begin{cases} i, & i - 1/3 < x \le i + 1/3, \\ 3x - 2i - 1, & i + 1/3 < x \le i + 2/3. \end{cases}$$

More generally, define $\Pi : \mathbb{R}^n \to \mathbb{R}^n$, by

$$\Pi(x) = [[x_1]_C, \ldots, [x_n]_C].$$

Then given any function $f : \mathbb{R}^n \to \mathbb{R}^m$, with $f(\mathbb{Z}^n) \subset \mathbb{Z}^m$, we can define a "robust version" by using the function $f \circ \Pi$.

Thus, given $[A, \mathbb{Z}_+, F]$, $A \subset \mathbb{Z}^n$, its transition function may be extended to a continuous function from $\mathbb{R}^n$ to $\mathbb{R}^n$ which is constant in a neighborhood of each point in A. Such a remark is actually a byproduct of a more general result needed below [113, p. 216]:

Fact 7.7 *Any continuous function $f : A \to \mathbb{R}^m$, A a closed subset of $\mathbb{R}^n$, may be extended to a continuous map $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^m$.*

Throughout the rest of this section we use continuous extensions as in the fact above, the notation $\tilde{f}$ always denoting such an extension of f.

§7.3.2 THE POWER OF EXACT CLOCKS

Later, we find that the ability to implement precise timing pulses is a strong system characteristic, enabling one to implement equations with powerful simulation capabilities. To this end, define

Definition 7.8 (Exact Clock) *A function $S : \mathbb{R}_+ \to \mathbb{Z}$ is an **exact m-ary clock with pulse-width T** or simply (m, T)-clock if*

1. *It is piecewise continuous with finite image $Q = \{0, \dots, m-1\}$, $m \geq 2$.*

2. *For all $t \in (kT, (k+1)T)$, $S(t) = i$ if $k \equiv i \pmod m$.*

All reviewed hybrid systems models, WHS, TDA, BGM, NKSD, ASL, and BAUT, can implement (m, T)-clocks, as the results of §3.10 and the following shows.

Example 7.9 1. *The BAUT model implements (m, T)-clocks: Choose $Z = \{0, \dots, m-1\}$ and*

$$\dot{p} = 1/T, \qquad\qquad p(0) = 0,$$
$$z\lceil p \rceil = (z\lfloor p \rfloor + 1) \bmod m, \qquad z\lfloor 0 \rfloor = 0.$$

Then $S(t) = z\lfloor p(t) \rfloor$ is an (m, T)-clock.

2. *The TDA model implements (m, T)-clocks: Set $p = m$ if m even, $p = m+1$ if m odd. Choose state space $\mathbb{R} \times Q$, $Q = \{0, \dots, p-1\}$. Define the continuous dynamics as*

$$f(x, q) = c_q(-1)^q,$$

$q \in Q$. Set $c_q = 1$ for all q if m is even; set $c_q = 1$ for $q \in \{0, \dots, m-2\}$, $c_q = 2$ for $q \in \{m-1, m\}$, if m odd. In each case define the switching manifolds by

$$g_{2k,2k+1}(x) = x - T,$$
$$g_{2k+1,(2k+2)\bmod p}(x) = -x.$$

Setting $x(0) = 0$, $S(t) = q(t)$ and

$$S(t) = q(t) - (m-1)[q(t)/3(m-1)]_C$$

are (m, T)-clocks when m is even and odd, respectively.

As an example of the simulation power one obtains with access to an exact clock, consider the following:

Theorem 7.10 *Every reversible discrete dynamical system F defined on a closed subset of $\mathbb{R}^n$ can be S-simulated by a system of continuous ODEs in $\mathbb{R}^{2n}$ with a $(2, T)$-clock, S, as input.*

Proof.

$$\dot{x}(t) = T^{-1}[\tilde{G}(z) - z](1 - S(t)),$$
$$\dot{z}(t) = T^{-1}[x - \tilde{H}(x)]S(t),$$

where $\tilde{G}$ and $\tilde{H}$ are continuous extensions of $G = F(\cdot, 1)$ and $H = F(\cdot, -1)$, respectively. Starting this system at $t = 0$ with $x(0) = z(0) = x_0$, $x_0 \in$ domain G, one sees that $x(2kT) = z(2kT) = G^k(x_0)$. Here, $\psi(x, z) = x$ for $x = z$, $x \in$ domain G. □

This theorem shows that exact clocks allow one to S-simulate arbitrary reversible discrete dynamical systems on closed subsets of $\mathbb{R}^n$ with a system of ODEs in $\mathbb{R}^{2n}$. The idea of turning on and off separate systems of differential equations is key to the simulation. The effect of the simulation is that on alternating segments of time one "computes" the next state, then copies it, respectively. Then, the process is repeated. One readily sees that the exact way the continuous extensions in the proof are performed is not important.

As seen above each of the reviewed hybrid systems models can implement $(2, T)$-clocks. In particular, they can implement a $(2, T)$-clock with just a single ODE. Thus the simulations of the theorem can be performed with continuous state space $\mathbb{R}^{2n+1}$ in each of these cases. Further, they each require only 2 discrete states.

The generality of Theorem 7.10 allows us to conclude

Corollary 7.11 *Using S-simulation, any hybrid systems model that implements continuous ODEs and a $(2, T)$-clock has the power of universal computation.*

Proof. Using constructions as in Proposition 7.4, construct a reversible discrete dynamical system in $\mathbb{Z}^n$ equivalent to a universal, reversible TM (one whose transition function is invertible) [13, 138]. In turn, simulate it using the theorem. □

However, we want to explore simulation of non-reversible finite and infinite computational machines with hybrid and continuous dynamical systems. First, we show that the ability to set parameters on clock edges is strong.

Theorem 7.12 *Every discrete dynamical system F defined on a closed subset of $\mathbb{R}^n$ can be S-simulated by a system of continuous ODEs on $\mathbb{R}^{2n}$ (resp. $\mathbb{R}^n$) with a $(2, T)$-clock, S, as input and the ability to set parameters on clock edges.*

Proof. Define $G \equiv F(\cdot, 1)$. Both systems are initialized at $t = 0$ with $c = x(0) = x_0$, $x_0 \in$ domain G.

1. Initialize $z(0) = x_0$. Use

$$\dot{x}(t) = T^{-1}[\tilde{G}(z) - z](1 - S(t)),$$
$$\dot{z}(t) = T^{-1}[x - c]S(t).$$

The constant c is set to z when $t = kT$, k odd. One sees that $x(2kT) = z(2kT) = G^k(x_0)$. Choose $\psi(x, z) = x$ for $x = z$, $x \in$ domain G.

2. Use

$$\dot{x}(t) = T^{-1}[\tilde{G}(c) - c](1 - S(t)).$$

The constant c is set to x when $t = kT$, k even. One sees that $x(2kT) = G^k(x_0)$. Choose $\psi(x) = x$, $x \in$ domain G. □

Note that if F is not reversible, forward trajectories of the above systems of equations may merge. This situation is allowed by our definitions. The simplest example of this is $[\{0,1\}, \mathbb{Z}_+, F]$ with $F(0,1) = F(1,1) = 0$.

Corollary 7.13 *Any hybrid systems model that implements continuous ODEs, a $(2,T)$-clock, and setting parameters on clock edges, can S-simulate any TM, PDA, or FA; and, using S-simulation, has the power of universal computation.*

Proof. Combine the theorem and Proposition 7.4. □

In particular, the BGM model has this power (by defining the appropriate transition functions on the switching boundaries of the TDA $(2,T)$-clock given above).

§7.3.3 SIMULATION WITHOUT EXACT CLOCKS

Without an exact clock, one's simulation power is limited. However, one can still simulate discrete dynamical systems defined on arbitrary subsets of $\mathbb{Z}^n$. Next, we proceed to explicitly show that all hybrid systems models of §3 can simulate any discrete dynamical system on $\mathbb{Z}^n$. Indeed, we show that continuous ODEs can simulate them.

In the previous section we used an exact $(2,T)$-clock to precisely switch between two different vector fields in order to simulate discrete dynamical systems in $\mathbb{R}^n$. Again, the essential idea behind the simulations in this section is to alternately switch between two different vector fields. However, since we are simulating systems in $\mathbb{Z}^n$, using "robust versions" of their transition functions, and choosing well-behaved ODEs, it is not necessary to precisely time these switches using an exact clock. Indeed, we can use continuous functions to switch among vector fields.

It is still convenient to ensure, however, that only one vector field is active (non-zero) at any given time. Thus, we would like

Definition 7.14 (Inexact Clock) *An* **inexact (m,T)-clock**, $m \geq 2$, *is a continuous function* $S : \mathbb{R}_+ \to [0,1]^m$ *such that on each interval* $t \in [kT, (k+1)T]$ *with* $k \equiv i \pmod{m}$ *the following hold:* $S_{j+1}(t) = 0$, $0 \leq j \leq m-1$, $j \neq i$; $S_{i+1}(t) \equiv 1$ *on a sub-interval of length greater than or equal to* $T/2$.

It is also reasonable to require that transitions between 0 and 1 take place quickly or that there be some minimum separation between the times when $S_i > 0$, $S_j > 0$, $i \neq j$. Below, we need an inexact $(2,T)$-clock with the latter property.

What is key is that such inexact clocks do not require discontinuous vector fields, discontinuous functions, or discrete dynamics. They can be implemented as follows.

Example 7.15 (Inexact $(2,T)$-clock) *Define* $\dot{\tau}(t) = 1/T$, *initialized at* $\tau(0) = 0$. *Now, define*

$$S_{1,2}(\tau) = h_{\pm}[\sin(\pi\tau)],$$

where

$$h_+(r) = \begin{cases} 0, & r \leq \delta/2, \\ 2r/\delta - 1, & \delta/2 < r \leq \delta, \\ 1, & \delta < r, \end{cases}$$

$h_-(r) = h_+(-r)$, *and* $0 < \delta < \sqrt{2}/2$.

Thus, one can switch between two different systems of ODEs with (Lipschitz) continuous functions of the state of another (Lipschitz) ODE. This is why $2n+1$ dimensional ODEs are used below to simulate an n-dimensional discrete dynamical system.

We also need the following technical definitions:

Definition 7.16 (Non-degeneracy, Finite Gain) *A function* $f : \mathbb{R}^n \to \mathbb{R}^n$, *is non-degenerate* (resp. **finite gain**) *if there exist constants* $\beta \geq 0$, $M \geq 0$, *such that*

$$\|x\| \leq M\|f(x)\| + \beta, \qquad (\text{resp. } \|f(x)\| \leq M\|x\| + \beta),$$

for all $x \in X$.

Now we are ready for our main simulation result:

Theorem 7.17 *Every discrete dynamical system F defined on $Y \subset \mathbb{Z}^n$*

1. *can be SI-simulated by a system of continuous ODEs in $\mathbb{R}^{2n+1}$.*

2. *such that $F(\cdot, 1)$ is finite gain and non-degenerate can be I-simulated by a system of continuous ODEs in $\mathbb{R}^{2n+1}$.*

3. *such that Y is bounded can be SI-simulated by a system of Lipschitz ODEs in $\mathbb{R}^{2n+1}$.*

Proof. Let $G \equiv F(\cdot, 1)$ and $0 < \epsilon < 1/3$. $S_{1,2}$ and δ are as in the preceding example. For each $y \in Y$, define the set

$$H_y = \{(x, z, \tau) \mid \|x - y\|_\infty < \epsilon, \|z - y\|_\infty < \epsilon, \sin(\pi\tau) < \delta/2, \tau \bmod_C 2 < 1/2\},$$

Set $\psi(x, z, \tau) = \Pi(z) = y$ if $(x, z, \tau) \in H_y$. Note: the $\psi^{-1}(y) = H_y$ are open and disjoint. Initialize $x(0), z(0), \tau(0)$ in $\psi^{-1}(y)$, $y \in Y$.

1. Choose

$$\begin{aligned}
\dot{x} &= -\epsilon^{-2}[x - \tilde{G}(\Pi(z))]^3 S_1(\tau), \\
\dot{z} &= -\epsilon^{-2}[z - \Pi(x)]^3 S_2(\tau), \\
\dot{\tau} &= 1.
\end{aligned}$$

It is straightforward to verify $\Pi(z(2k)) = G^k(y)$, $k \in \mathbb{Z}_+$, and the interval constraint.

2. Let α and L be the finite gain, and β and M the non-degeneracy constants of G under norm $\|\cdot\|_\infty$. Choose

$$\begin{aligned}
\dot{x} &= -2\epsilon^{-1}[x - \tilde{G}(\Pi(z))]S_1(\tau), \\
\dot{z} &= -2\epsilon^{-1}[z - \Pi(x)]S_2(\tau), \\
\dot{\tau} &= 1/[1 + (L+1)\|z\|_\infty + \alpha + (M+1)\|x\|_\infty + \beta].
\end{aligned} \tag{7.1}$$

It is straightforward to verify $\Pi(z(t)) = G^k(y)$ on an interval about the time t_k where $\tau(t_k) = 2k$, $k \in \mathbb{Z}_+$.

3. Let $\beta = \max\{\|i - j\|_\infty \mid i, j \in Y\}$. Choose

$$\begin{aligned}
\dot{x} &= -2\beta\epsilon^{-1}[x - \tilde{G}(\Pi(z))]S_1(\tau), \\
\dot{z} &= -2\beta\epsilon^{-1}[z - \Pi(x)]S_2(\tau), \\
\dot{\tau} &= 1.
\end{aligned} \tag{7.2}$$

It is straightforward to verify $\Pi(z(2k)) = G^k(y)$, $k \in \mathbb{Z}_+$, and the interval constraint.

$\square$

Note that non-degeneracy and finite gain of the extension $\tilde{G}$ need not hold for points not in Y. Note also that the simulations above are "robust" in the sense that there is a neighborhood of initial conditions leading to the correct simulations. The import of part 2 of the theorem is that if $G \equiv F(\cdot, 1)$ is non-degenerate and may be extended to a Lipschitz function, then the ODEs used in the I-simulation are also Lipschitz.

Note also that the theorem continues to hold for any discrete dynamical system defined on $Y \subset \mathbb{R}^n$ such that there is some minimum separation between any two distinct points of Y.

The discrete dynamical systems equivalent to TMs given by Proposition 7.4 have transition functions that are both finite gain and non-degenerate. Unfortunately, the transition functions of systems equivalent even to PDA need not be Lipschitz. Consider a PDA which pushes a tape symbol γ on input symbol i_1 and pops γ on input symbol i_2 and test with inputs of the form i_1^{n+1} and $i_1^n i_2$. One may check that the "prime encodings" mentioned earlier lead to transition functions that are neither finite gain nor non-degenerate.

Nevertheless, relating the theorem back to simulation of TMs, PDA, and FA, we have many results, the most striking of which are:

Corollary 7.18 *Every TM, PDA, and FA can be SI-simulated by a system of continuous ODEs in $\mathbb{R}^3$.*

Every FA (resp. inputless FA) can be I-simulated (resp. SI-simulated) by a system of Lipschitz continuous ODEs in $\mathbb{R}^3$.

Using SI-simulation, there is a system of continuous ODEs in $\mathbb{R}^3$ with the power of universal computation.

Proof. Everything is immediate from the theorem and Propositions 7.4 and 7.5 except that the FA transition function is Lipschitz, which is readily checked. $\square$

Of course, any hybrid systems model that implements continuous (resp. Lipschitz) ODEs has similar powers. In particular, the models reviewed in §3 do.

Finally, all the simulation results for discrete dynamical systems on $\mathbb{Z}$ can be extended from continuous to smooth vector fields by using C^∞ interpolation (with so-called "bump" functions [59]) rather than linear interpolation in extending their transition functions and the functions $[\cdot]_C$ and $h_\pm$, and by replacing $\|\cdot\|_\infty$ with $\|\cdot\|_2$ in Equation (7.1).

§7.4 IMPLEMENTING ARBITERS

In this section, we contrast the capabilities of hybrid and continuous dynamical systems by using the famous asynchronous arbiter problem [26, 100, 144].

We begin in the first subsection with a discussion of the arbiter problem. Next, we prove that one cannot implement an asynchronous arbiter using a system of Lipschitz ODEs continuous in inputs and outputs, i.e., a system of the form of Equation (2.2) with f Lipschitz in x, continuous in u and h continuous [73, p. 297]. Finally, we show that all hybrid systems models in §3 can implement arbiters, even when their continuous dynamics is a system of Lipschitz ODEs continuous in inputs and outputs.

§7.4.1 THE ARBITER PROBLEM

The definition and technical specifications of an (asynchronous) arbiter below are adapted from [144].

An **arbiter** is a device that can be used to decide the winner of two-person races. It is housed in a box with two input buttons, labeled B_1 and B_2, and two output lines, W_1 and W_2, that can each be either 0 or 1. For ease of exposition, let the vectors

$$B = (B_1, B_2), \qquad W = (W_1, W_2)$$

denote the button states and outputs, respectively. There is also a reset button, R. Below, the buttons B_i, R are taken to be 1 when they are pressed, 0 when they are unpressed.

After the system has been reset, the output should be $(1, 0)$ if B_1 is pressed before B_2; it should be $(0, 1)$ if B_2 is pressed before B_1. Let T_i denote the time that button B_i is pressed. Then, the function of the arbiter is to make a binary choice based on the value of the continuous variable $T_1 - T_2$. If the difference is negative, the output should be $(1, 0)$; if it is positive, the output should be $(0, 1)$. Upon reset, the output is set to $(0, 0)$.

Here are the arbiter's technical specifications:

S1. Pressing the reset button, R, causes the output to become $(0, 0)$, perhaps after waiting for some specified time, denoted T_R, where it remains until one or both buttons are pressed.

S2. The pressing of either or both buttons B_i causes, after an interval of at most T_d units, the output to be either $(0, 1)$ or $(1, 0)$; the output level persists until the next reset input.

S3. If B_1 is pressed T_a seconds or more before B_2 is pressed, then the output is $(1, 0)$, indicating that B_1 was pressed first. Similarly, if B_2 is pressed T_a seconds or more before B_1 is pressed, then the output is $(0, 1)$, indicating that B_2 was pressed first.

S4. If B_1 and B_2 are pressed within T_a seconds of each other, then the output is either $(1, 0)$ or $(0, 1)$—one does not care which—after the T_d-second interval.

The arbiter problem is

Problem 7.19 (Asynchronous arbiter problem) *Build a device that meets the specifications* S1–S4.

§7.4.2 YOU CAN'T IMPLEMENT AN ARBITER WITH LIPSCHITZ ODEs

In this section, we show that it is impossible to build a device, described as a system of Lipschitz ODEs continuous in the required inputs and outputs, that implements the arbiter specifications.

First we give a generic system of Lipschitz ODEs with the required properties:

$$\begin{aligned} \dot{x}(t) &= f(x(t), B(t)), \\ W(t) &= h(x(t)), \end{aligned} \qquad (7.3)$$

where $x(t) \in \mathbb{R}^n$, $W(t) \in \mathbb{R}^2$, $B(t) \in \{0, 1\}^2$, with $B(\cdot)$ piecewise continuous. Each $f(\cdot, B)$, $B \in \{0, 1\}^2$, is Lipschitz. Thus, each vector field $f(\cdot, B)$ defines a continuous dynamical system $\phi_{(B_1, B_2)}$, with $\phi_{(B_1, B_2)}(x_0, T)$ the solution at time T of $\dot{x}(t) = f(x(t), B_1, B_2)$ starting at $x(0) = x_0$. Further, $h : \mathbb{R}^n \to \mathbb{R}^2$ is continuous. Note that the action of the reset button is unmodeled; it is not necessary to the proof, which assumes it remains unpressed on the interval of interest.

Since h is continuous, there exists a constant $\delta_{\sqrt{2}} > 0$ such that

$$\|h(x) - h(x')\|_2 < \sqrt{2} \qquad \text{whenever} \qquad \|x - x'\| < \delta_{\sqrt{2}}, \tag{7.4}$$

Define $L_W = \sqrt{2}/\delta_{\sqrt{2}}$.

Now, we are ready to settle the arbiter problem in this framework:

Theorem 7.20 *For no choice of the values for T_a and T_d is it possible to build a device described by Equation (7.3) that meets the arbiter specifications S1–S4.*

Proof. The proof is by contradiction, assuming there is a device described by Equation (7.3) which satisfies the specifications.

Assume that the arbiter has been reset, is in state $x(0) = x_0$ at time $t = 0$ with $h(x_0) = (0,0)$, and that one of the buttons is pressed at time $t = 0$. (This is without loss of generality as the equations are autonomous.) Also, assume that the reset button is not pressed until some time $T_R \gg T_a + T_d$.

The behavior of the device from $t = 0$ to $t = T_R$ is completely determined by which button was pressed first and at what time the second button is pressed (if ever). Therefore, let $x_\rho(t)$ denote the solution at time t of Equation (7.3) starting at time $t = 0$ at state $x(0) = x_0$ with fixed parameter $\rho \equiv T_1 - T_2$. Thus, ρ represents the difference between the times when B_1 and B_2 are pressed. If B_1 is pressed but B_2 is never pressed, set $\rho = -\infty$. If B_2 is pressed but B_1 is never pressed, set $\rho = \infty$.

The arbiter specifications require that for $T_a + T_d \leq t \leq T_R$,

$$h(x_\rho(t)) = \begin{cases} (1,0), & \rho \leq -T_a, \\ (0,1), & \rho \geq T_a, \\ (1,0) \text{ or } (0,1), & \text{otherwise.} \end{cases}$$

These specifications and Lemma 7.23 (in the Appendix) are such that for any $\delta > 0$, one can find $-T_a \leq \sigma < \tau \leq T_a$, with $\tau - \sigma < \delta$, and with one of $h(x_\sigma(T_a + T_d))$, $h(x_\tau(T_a + T_d))$ equal to $(1,0)$ and the other equal to $(0,1)$.

Pick $\delta < \min\{T_a, T_d, 1/L\}$, where $L > 0$ is a finite bound of the maximum of the four Lipschitz constants corresponding to each of the $f(\cdot, B)$. Define

$$c = \max\left\{\|f(x_0, 1, 0)\|, \|f(x_0, 0, 1)\|, \|f(x_0, 1, 1)\|\right\}.$$

Note $c > 0$, for otherwise $h(x_\sigma(t)) = h(x_\tau(t)) = h(x_0)$ for all $0 \leq t \leq T_R$, a contradiction.

For ease of notation, let F_t, G_t, H_t denote the fundamental solutions $\phi_{(0,1)}(\cdot, t)$, $\phi_{(1,0)}(\cdot, t)$, and $\phi_{(1,1)}(\cdot, t)$, respectively. Also, let $X_t = x_\sigma(t)$ and $Y_t = x_\tau(t)$. Note that $X_0 = Y_0 = x_0$.

The proof splits into three cases:

1. $0 \leq \sigma < \tau \leq T_a$.

2. $-T_a \leq \sigma < \tau \leq 0$.

3. $-T_a < \sigma < 0 < \tau < T_a$.

Case 1. In this case,

$$X_t = \begin{cases} F_t(x_0), & 0 \leq t \leq \sigma, \\ H_{t-\sigma}(F_\sigma(x_0)), & \sigma \leq t \leq T_R, \end{cases}$$

$$Y_t = \begin{cases} F_t(x_0), & 0 \le t \le \tau, \\ H_{t-\tau}(F_\tau(x_0)), & \tau \le t \le T_R. \end{cases}$$

Thus, $X_\sigma = Y_\sigma$. Now, by Corollary 7.25

$$\|Y_\tau - Y_\sigma\| \le cL^{-1}(e^{L\tau} - e^{L\sigma}).$$

Thus, Lemma 7.26 gives

$$\begin{aligned} \|X_{\sigma+T_d} - Y_{\tau+T_d}\| &\le cL^{-1}(e^{L(\tau-\sigma)} - 1)e^{L\sigma}e^{LT_d}, \\ &\le cL^{-1}(e^{L\delta} - 1)e^{L(T_a+T_d)}, \\ &\le c\delta(e-1)e^{L(T_a+T_d)}, \end{aligned}$$

where the last line follows from $L\delta < 1$. But by assumption,

$$\sqrt{2} = \|h(X_{\sigma+T_d}) - h(Y_{\tau+T_d})\|_2,$$

so that Equation (7.4) yields

$$K_1 \equiv \sqrt{2}\Big/\Big[cL_W(e-1)e^{L(T_a+T_d)}\Big] \le \delta.$$

Case 2. The argument is similar to Case 1 and yields the same inequality on δ.

Case 3. In this case,

$$X_t = \begin{cases} G_t(x_0), & 0 \le t \le |\sigma|, \\ H_{t-|\sigma|}(G_{|\sigma|}(x_0)), & |\sigma| \le t \le T_R, \end{cases}$$

$$Y_t = \begin{cases} F_t(x_0), & 0 \le t \le \tau, \\ H_{t-\tau}(F_\tau(x_0)), & \tau \le t \le T_R. \end{cases}$$

Note that $\max\{|\sigma|, |\tau|\} \le \delta$. This and Lemma 7.24 give

$$\left\|X_{|\sigma|} - Y_\tau\right\| \le \left\|X_{|\sigma|} - x_0\right\| + \|Y_\tau - x_0\| \le 2cL^{-1}(e^{L\delta} - 1)$$

Thus, Lemma 7.26 gives

$$\begin{aligned} \left\|X_{|\sigma|+T_d} - Y_{\tau+T_d}\right\| &\le 2cL^{-1}(e^{L\delta} - 1)e^{LT_d}, \\ &\le 2ce^{LT_d}(e-1)\delta, \end{aligned}$$

where the last line follows from $L\delta < 1$. But by assumption,

$$\sqrt{2} = \left\|h\left(X_{|\sigma|+T_d}\right) - h(Y_{\tau+T_d})\right\|_2,$$

so that Equation (7.4) yields

$$K_3 \equiv 1\Big/\Big[\sqrt{2}cL_W(e-1)e^{LT_d}\Big] \le \delta.$$

Thus, choosing $\delta < \min\{T_a, T_d, 1/L, K_1, K_3\}$, would have achieved a contradiction in all

three cases.   □

The basic argument used above is that one cannot have a continuous map from a connected space (e.g., $\mathbb{R}$ containing ρ) to a disconnected space (e.g., $\{(1,0), (0,1)\}$) [113]. Nevertheless, one must prove that the map given by the device is indeed continuous before one makes such an appeal. Above, we have explicitly demonstrated the continuity of the system of switched differential equations describing the arbiter.

§7.4.3   IMPLEMENTING ARBITERS WITH HYBRID SYSTEMS

In this section it is shown that each of the hybrid systems models can implement an arbiter. Given the results of §3.9, it is enough to implement one using the BAUT and TDA models. However, the problem is such that we must add inputs and outputs to these models, which is done in an obvious way.

In each case, the continuous dynamics is a system of Lipschitz ODEs continuous in inputs and outputs, the essential "resolving power" coming from the mechanisms implementing the discrete dynamics.

We first implement an arbiter with a hybrid system *à la* Brockett:

Proposition 7.21 *There exists a system of equations in the BAUT model with inputs and outputs that meets the arbiter specifications S1–S4.*

Proof. We design for $T_a = T_d/2 = T_m$.

$$\dot{x} = [2(4z\lfloor p \rfloor - 1)\max(B_1, B_2)T(x)/T_m](1 - R) - (2x/T_R)R,$$
$$\dot{p} = [2B_1(B_1 - B_2)(1 - z\lfloor p \rfloor)/T_m](1 - R) + (z\lfloor p \rfloor/T_R)R,$$
$$z\lceil p \rceil = (z\lfloor p \rfloor + 1) \bmod 2,$$
$$W = h(x),$$

where

$$h(x) = \begin{cases} (0,1), & x \leq -3, \\ (0, |x| - 2), & -3 \leq x \leq -2, \\ (0,0), & -2 \leq x \leq 2, \\ (x - 2, 0), & 2 \leq x \leq 3, \\ (1,0), & 3 \leq x, \end{cases}$$

$$T(x) = \begin{cases} 1, & |x| \leq 4, \\ 5 - |x|, & 4 \leq |x| \leq 5, \\ 0, & 5 \leq |x|. \end{cases}$$

Let's examine these equations when B_2 is pressed at time $t = 0$. Let $T_1 > 0$ denote the time at which B_1 is pressed. The equations are assumed to be properly reset so that without loss of generality, we assume that $|x(0)| < 1$ and $p(0) \in [2k, 2k + 1)$, for some $k \in \mathbb{Z}_+$, and $z\lfloor p(0) \rfloor = 0$. Also, we assume that the reset button is inactive ($R = 0$) from $t = 0$ to $t = t_R > 2T_m$. In this case, the two equations are simply (no matter when B_1 is pressed)

$$\dot{x} = -2T(x)/T_m,$$
$$\dot{p} = 0,$$

so that $x(t) < -3$ and hence $W(t) = (0,1)$ for $t \in [2T_m, t_R]$.

Now, we look at these equations under the same assumptions, excepting B_1 is pressed

at $t = 0$ and B_2 is pressed at $t = T_2 > 0$. Now there are two cases: $T_2 < t_z$ and $T_2 \geq t_z$, where $t_z = [1 - (p - \lfloor p \rfloor)]T_m/2 \leq T_m/2$ is the time when $z(\lfloor p(t) \rfloor)$ would first equal 1 if B_2 were not pressed before it. In the second case, by time t_z the equations are

$$\dot{x} = 6T(x)/T_m,$$
$$\dot{p} = 0,$$

so that $x(t) > 4$ and hence $W(t) = (1, 0)$ for $t \in [t_z + T_m, t_R] \supset [2T_m, t_R]$. In the first case, the first equation remains

$$\dot{x} = -2T(x)/T_m,$$

so that $x(t) < -3$ and hence $W(t) = (0, 1)$ for $t \in [2T_m, t_R]$. $\square$

The reset behavior is readily verified.

Now, we implement an arbiter with TDA:

Proposition 7.22 *There exists a system of equations in the TDA model with inputs and outputs that meets the arbiter specifications S1–S4.*

Proof.   For convenience, define $T_m = \min\{T_d, T_a\}$. Define the continuous dynamics, $f(x, q, B_1, B_2, R)$, which depends on states $x \in \mathbb{R}^2$, $q \in \{1, 2, 3\}$, and inputs B_1, B_2, and R, each in $\{0, 1\}$, as follows:

$$
\begin{aligned}
f(x, 1; B_1, B_2, 0) &= (B_1[B_1 - B_2], B_2), \\
f(x, 2; \cdot, \cdot, 0) &= (0, 0), \\
f(x, 3; \cdot, \cdot, 0) &= (0, 0), \\
f(x, \cdot; \cdot, \cdot, 1) &= -\frac{T_m}{\epsilon T_R} x,
\end{aligned}
$$

with switching boundaries defined as follows:

$$
\begin{aligned}
g_{1,2}(x) &= 4\epsilon^2 - \|x - (T_m, 0)\|_2^2, \\
g_{1,3}(x) &= x_2 - T_m, \\
g_{2,1}(x) = g_{3,1}(x) &= \epsilon^2 - \|x\|_2^2,
\end{aligned}
$$

where $0 < \epsilon < T_m/4$. Finally, define the output $W = h(x)$ where

$$
h(x) = \begin{cases}
(1, 0), & x_2 \leq T_m/2, \\
(1 - 4(x_2/T_m - 1/2), 4(x_2/T_m - 1/2)), & T_m/2 < x_2 < 3T_m/4, \\
(0, 1), & 3T_m/4 \leq x_2.
\end{cases}
$$

One readily verifies that it behaves correctly. $\square$

§7.5 DISCUSSION

We now turn to some discussion. Our simulation of arbitrary Turing machines imply that, in general, questions regarding the dynamical behavior of hybrid systems with continuous ODEs—and even well-behaved ODEs themselves—are computationally undecidable. See [107, 108] for a discussion of such questions. Further, the ODEs simulating these machines may be taken smooth and do not require the machines to be reversible (cf. [108, p. 228]).

The import of S-simulation here is that such simulations take only "linear time" [49]. The import of I-simulation is that the readout times for which the state/tape is valid are non-empty intervals. Indeed, the intervals are at least some minimum length. Also, the simulations were "robust" in the sense that they can tolerate small errors in the coding of the initial conditions. Though not required by our definitions, these contained balls of at least some minimum diameter.

The explicit formulation and solution of the asynchronous arbiter problem in an ODE framework appears to be new. One should note that in our ODE model, the inputs B_i were assumed to be ideal in the sense that they switch from 0 to 1 instantaneously. Imposing continuity assumptions on B as signals in $[0, 1]^2$ leads to a similar result.

To demonstrate the computational capabilities of hybrid and continuous dynamical systems summarized above, we constructed low-dimensional discrete dynamical systems in $\mathbb{Z}^n$ equivalent to Turing machines (TMs), pushdown automata (PDA), and finite automata (FA). It is well-known that certain discrete dynamical systems are equivalent to TMs and possess the power of universal computation (see, e.g., [49, 107, 131]). Our systems were constructed with the goal of simulation by continuous/Lipschitz ODEs in mind. One notes that while it is perhaps a trivial observation that there are systems of (Lipschitz) ODEs with the power of universal computation—just write down the ODEs modeling your personal computer—this requires a system of ODEs with a potentially infinite number of states.

The best definition of "simulation" is not apparent. While stated in terms of our definitions of simulation, the simulation results of §7.3 are intuitive and would probably continue to hold under alternate definitions of simulation.

Related to our general simulation results is a theorem by N. P. Zhidkov [157] (see also [129, p. 135]), that states if a reversible discrete dynamical system is defined on a compact subset $K \subset \mathbb{R}^n$, then there exists on a subset of $\mathbb{R}^{2n+1}$ a reversible continuous dynamical system that is defined by ODEs and has K as a global section.

It is possible to take a different approach than the one in §7.3 and construct smooth systems of ODEs with inputs that "simulate" finite automata. For instance, in [36] Brockett used a system of his so-called double-bracket equations (also see [37]) to "simulate" the step-by-step behavior of a FA. This was done by coding the input symbols of the FA in a function of time that is the "control input" to a system of double-bracket equations. Specifically, if the input alphabet is $I = \{u_1, \ldots, u_m\}$, the input string $u_{i_0}, u_{i_1}, u_{i_2}, \ldots$ is encoded in a time function, $u(t)$, that is i_k on the intervals $[2kT, (2k+1)T]$ and zero otherwise. In this paper, we encoded the full input string in the initial condition of our simulations.

In [36], Brockett was interested in the capabilities of his double-bracket equations. However, the resulting "simulations" of FA happen to behave poorly with respect to our definitions of simulation. Nevertheless, the key idea of his simulations of FA is that the input coding, $u(t)$, is used in such a way that it alternately switches between two different systems of double-bracket equations. This idea is critical in our simulations of discrete dynamical systems with ODEs.

It is not hard to see that one could use the same approach as that in [36] but more well-behaved systems of ODEs to simulate the step-by-step behavior of FA. Consider a FA with transition function δ, states $Q = \{q_1, \ldots, q_n\}$, and input alphabet I as above. Code state q_i as i and consider the first two equations of Equation (7.2). Choose $\beta = n$ and replace, respectively, S_1, S_2, and G with $h_+(u(t))$, $h_-(u(t) - 1)$, and

$$D : \{1, \ldots, n\} \times \{1, \ldots, m\} \to \{1, \ldots, n\},$$

defined by $D(i,j) = k$ if $\delta(q_i, u_j) = q_k$. The result is that any FA may be SI-simulated by a system of ODEs in $\mathbb{R}^2$ with input. This was also announced in [22].

In [7], it is shown that so-called **piecewise-constant derivative systems (PCDs)** in $\mathbb{R}^3$ can "simulate" arbitrary inputless FA, inputless PDA, and TMs. Briefly, the notion of simulation used is that of I-simulation excepting as follows. First, the intervals in T can be open, closed, or half-closed; '$\leq$' may replace '$<$' in the constraints on τ_k, τ_k'; and there is no ϵ constraint. Also, there is no continuity constraint on ψ and for each $y \in Y$ there need exist only one point in $\psi^{-1}(y)$ for which the equation holds. However, there is the constraint that each $\psi^{-1}(y)$ is convex and relatively-open (i.e., open in the subspace of its affine hull). For convenience, we refer to this notion as AM-simulation. Since our I-simulations in Theorem 7.17 had $\psi^{-1}(y)$ open and convex, they are AM-simulations (here we are thinking of τ in $\mathbb{R}/2\mathbb{Z}$, or in a circle embedded in $\mathbb{R}^2$, with appropriate changes).

Convexity of $\psi^{-1}(y)$ may be a desirable property. For instance, it excludes simulation of FA by "unraveling" their transition diagrams into trees, a simple example of which is recounted in [7]. On the other hand, consider the case of a universal TM, U, simulating an inputless FA, A. Certainly, there could be many distinct configurations of U in which the current state of A is written on, say, its first tape cell. Then, even if the inverse images of the configurations of U are convex, the inverse images of the valid configurations with, say, q in the first tape cell need not be, preventing indirect AM-simulation of A through AM-simulation of U. In any case, we could have added the constraint that each $\psi^{-1}(y)$ be convex to our definitions of simulation with little change in any of our results.

Finally, in [7] Asarin and Maler use three-dimensional PCDs to AM-simulate inputless FA. They also point out that three dimensions are necessary in order to AM-simulate, with autonomous ODEs, inputless FA whose transition graphs are not planar. While their argument is fine, the transition graphs of deterministic inputless FA are always planar and it is straightforward to construct PCDs (and continuous ODEs) in two dimensions that AM-simulate such FA. Moreover, even though the transition graphs of FA (with inputs) need not be planar, their argument does not contradict the result in $\mathbb{R}^2$ derived in this section, since it uses non-autonomous ODEs.

§7.6 NOTES

The work in this chapter appeared in [31]. Our simulation of arbitrary Turing machines was first announced in [22].

Prof. John Wyatt presented the arbiter problem and the challenge to produce an ODE-based model and proof in his nonlinear systems course at MIT [153]. The explicit formulation and solution of the asynchronous arbiter problem in an ODE framework is excerpted from [26], which also discusses bounds on the performance of systems approximating arbiter behavior, arising from the explicit proof. Specifically, while the proof prohibits the construction of an arbiter with $T_d = O(1)$, it does not prohibit an arbitration device with $T_d = O(\ln(1/\rho))$. Such a device is given in [26].

Dr. Charles Rockland has recently brought to our attention an interesting example of the complexity that can arise in low-dimensional differential equations. In particular, [125] gives a non-trivial fourth-order algebraic differential equation (given by a polynomial with integer coefficients) exhibiting smooth solutions dense in $C(\mathbb{R}, \mathbb{R})$.

§7.7 APPENDIX

Lemma 7.23 *If X is a connected metric space, Y is a discrete topological space with two points, and $f : X \to Y$ is surjective, then for every $\delta > 0$ one can find $x, z \in X$ such that $d(x, z) < \delta$ and $f(x) \neq f(z)$.*

Proof. Assume the contrary. Then for all $x \in X$, $f(B_\delta(x)) = \{f(x)\} \subset V$, where $B_\delta(x)$ denotes the ball of radius δ about x and V is any open set about $f(x)$ in Y. Thus, f is continuous [113]. But f continuous and X connected implies $f(X) = Y$ is connected [113], a contradiction. $\qquad\square$

Lemma 7.24 *Suppose $\dot{x}(t) = f(x(t))$ with f globally Lipschitz continuous in x with constant $L_f \geq 0$. Then, for any L such that $L \geq L_f$ and $L > 0$, and any $t_2 \geq t_1$,*

$$\|x_{t_2} - x_{t_1}\| \leq \|f(x_{t_1})\| L^{-1}(e^{L(t_2 - t_1)} - 1).$$

Proof. Note that for $t \geq t_1$,

$$x_t - x_{t_1} = \int_{t_1}^t f(x_{t_1})ds + \int_{t_1}^t [f(x_s) - f(x_{t_1})]\, ds.$$

So that

$$\begin{aligned}
\|x_t - x_{t_1}\| &\leq \int_{t_1}^t \|f(x_{t_1})\| ds + \int_{t_1}^t \|f(x_s) - f(x_{t_1})\|\, ds \\
&\leq (t - t_1)\|f(x_{t_1})\| + \int_{t_1}^t L\|x_s - x_{t_1}\|\, ds.
\end{aligned}$$

Now, substituting $\tau = t - t_1$ and $\sigma = s - t_1$, this becomes

$$\|x_{\tau + t_1} - x_{t_1}\| \leq \tau\|f(x_{t_1})\| + \int_0^\tau L\|x_{\sigma + t_1} - x_{t_1}\|\, d\sigma$$

Finally, defining $u(\tau) = \|x_{\tau + t_1} - x_{t_1}\|$, this becomes

$$u(\tau) \leq \tau\|f(x_{t_1})\| + \int_0^\tau Lu(\sigma)\, d\sigma.$$

The result now follows from the well-known Bellman-Gronwall inequality [58, p. 252]. $\qquad\square$

Corollary 7.25 *Under the same assumptions plus the fact that the system was in state x_{t_0} at time $t_0 \leq t_1 \leq t_2$,*

$$\|x_{t_2} - x_{t_1}\| \leq \|f(x_{t_0})\| L^{-1}(e^{L(t_2 - t_0)} - e^{L(t_1 - t_0)}).$$

Proof. Note that Lipschitz continuity gives

$$\|f(x_{t_1})\| \leq L\|x_{t_1} - x_{t_0}\| + \|f(x_{t_0})\|.$$

But, the lemma gives in turn

$$\|x_{t_1} - x_{t_0}\| \leq \|f(x_{t_0})\| L^{-1}(e^{L(t_1 - t_0)} - 1).$$

So that the result follows. □

The following lemma is well-known (see, e.g., [73, p. 169]).

Lemma 7.26 *Let $y(t), z(t)$ be solutions to $\dot{x}(t) = f(x(t))$ where f has global Lipschitz constant $L \geq 0$. Then for all $t \geq t_0$,*

$$\|y(t) - z(t)\| \leq \|y(t_0) - z(t_0)\|e^{L(t-t_0)}.$$

□

Chapter 8

Analysis Tools

§§8.2–8.4 introduce some analysis tools for continuous switching systems. We prove theorems regarding limit cycles and robustness of such systems. The remainder of the chapter outlines some work on the stability analysis of switched and hybrid systems. We introduce *multiple Lyapunov functions* as a tool for analyzing Lyapunov stability and use *iterated function systems (IFS)* theory as a tool for Lagrange stability. We also discuss the case where the switched systems are indexed by an arbitrary compact set.

§8.1 INTRODUCTION

In the first part of the chapter, we develop general tools for analyzing continuous switching systems. For instance, we prove an extension of Bendixson's Theorem to the case of Lipschitz continuous vector fields. This gives us a tool for analyzing the existence of limit cycles of continuous switching systems. We also prove a lemma dealing with the continuity of differential equations with respect to perturbations that preserve a linear part. Colloquially, this lemma demonstrates the robustness of ODEs with a linear part. For purpose of discussion, we call it the *Linear Robustness Lemma*. This lemma is useful in easily deriving some of the common robustness results of nonlinear ODE theory (as given in, for instance, [11]). This lemma also becomes useful in studying singular perturbations if the fast dynamics are such that they maintain the corresponding algebraic equation to within a small deviation. We give some simple propositions that allow us to do this type of analysis.

The extension of Bendixson's Theorem and the Linear Robustness Lemma have uses beyond those explicitly espoused here and should be of general interest to systems theorists.

§8.5 introduces "multiple Lyapunov functions" as a tool for analyzing Lyapunov stability of switched systems. In §8.6 iterative function systems are presented as a tool for proving Lagrange stability and positive invariance. We also address the case where $\{1,\ldots,N\}$ in Equations (4.6) and (4.7) is replaced by an arbitrary compact set. We conclude with some discussion.

Appendix A collects some tedious proofs. Appendix B treats the background, statement, and proof of our extension of Bendixson's Theorem.

In the next chapter, we use the above tools to analyze some example continuous switching systems motivated by a realistic aircraft control problem.

§8.2 EXISTENCE OF LIMIT CYCLES

Suppose we are interested in the existence of limit cycles of continuous switching systems in the plane. The traditional tool for such analysis is Bendixson's Theorem. But under our

model, systems typically admit vector fields that are Lipschitz, with no other smoothness assumptions. Bendixson's Theorem, as it is traditionally stated (e.g., [67, 142]), requires continuously differentiable vector fields and is thus not of use in general. Therefore, we offer an extension of Bendixson's Theorem to the more general case of Lipschitz continuous vector fields. Its proof is based on results in geometric measure theory (which are discussed in Appendix 8.10).

Theorem 8.1 (Extension of Bendixson's Theorem) *Suppose D is a simply connected domain in $\mathbb{R}^2$ and $f(x)$ is a Lipschitz continuous vector field on D such that the quantity $\nabla f(x)$ (the **divergence** of f, which exists almost everywhere) defined by*

$$\nabla f(x) = \frac{\partial f_1}{\partial x_1}(x_1, x_2) + \frac{\partial f_2}{\partial x_2}(x_1, x_2)$$

is not zero almost everywhere over any subregion of D and is of the same sign almost everywhere in D. Then D contains no closed trajectories of

$$\dot{x}_1(t) = f_1[x_1(t), x_2(t)], \tag{8.1}$$
$$\dot{x}_2(t) = f_2[x_1(t), x_2(t)]. \tag{8.2}$$

Proof. Similar to that of Bendixson's Theorem [142, pp. 31–32] after using an extension of the divergence theorem known as the Gauss-Green-Federer Theorem [109, pp. 114–115]. (See Appendix 8.10.)   □

Finally, we give an example which shows the necessity of Lipschitz continuity of the vector fields.

Example 8.2 *Consider Example 8.9. Note that if the roles of A and B are interchanged, then the resulting system is asymptotically stable. Thus, continuity of solutions and the intermediate value theorem imply that there exists $\lambda \in (0,1)$ such that $f_1 = \lambda B + (1 - \lambda)A$ and $f_2 = \lambda A + (1 - \lambda)B$ results in a closed trajectory. Yet, $\nabla f_1 < 0$ and $\nabla f_2 < 0$.*

§8.3 ROBUSTNESS OF ODEs

In this subsection, we summarize some results that show the robustness of solutions of ordinary differential equations with respect to perturbations of the vector field. First, we give and prove a basic lemma in ODE theory that demonstrates robustness of solutions to arbitrary perturbations. Then, we consider perturbations that preserve a linear part. This allows us to obtain more useful bounds. We call the result the Linear Robustness Lemma.

The proofs of both lemmas depend critically on the well-known Bellman-Gronwall inequality [58, p. 252], which is reprinted in Appendix 8.9 for convenience. The first is a basic lemma in ODE theory that was given without proof in [153]. It is useful as a comparison with our new result, Lemma 8.4. For completeness, we furnish a proof in Appendix 8.9.[1]

Lemma 8.3 *Given*

$$\dot{x} = F(x, t), \quad x(0) = x_0,$$
$$\dot{y} = G(y, t), \quad y(0) = x_0.$$

---

[1]Proofs that interrupt discussion flow or do not use novel techniques are relegated to Appendix 8.9.

Suppose that F is globally Lipschitz continuous and "close to G," i.e.,

$$\|F(x,t) - F(y,t)\| \le L\|x - y\|, \quad \text{for all } x, y, t,$$
$$\|F(x,t) - G(x,t)\| \le \epsilon, \quad \text{for all } x, t.$$

Then if $L \ne 0$

$$\|x(t) - y(t)\| \le \frac{\epsilon}{L}\left(e^{Lt} - 1\right), \quad \text{for all } t \ge 0.$$

If $L = 0$, then $\|x(t) - y(t)\| \le \epsilon t$.

Proof. (See Appendix 8.9.) □

The problem with this result is that (except in the trivial case) $L > 0$, so the bound diverges exponentially. Thus it is not useful in deducing stability of a nearby system, nor in examining robustness of a well-behaved model to perturbations in the vector field. There are some tools for this in the literature, under the heading "stability under persistent disturbances." For example, [127, p. 72] gives a local result. We are more interested in what one can say globally. Along these lines we consider perturbations that preserve a well-defined portion of the dynamics, a linear part. Here is our main result:

Lemma 8.4 (Linear Robustness Lemma) *Given*

$$\dot{x} = Ax + F(x,t), \quad x(0) = x_0,$$
$$\dot{y} = Ay + G(y,t), \quad y(0) = x_0.$$

Suppose that F is globally Lipschitz continuous and "close to G," i.e.,

$$\|F(x,t) - F(y,t)\| \le L\|x - y\|, \quad \text{for all } x, y, t,$$
$$\|F(x,t) - G(x,t)\| \le \epsilon, \quad \text{for all } x, t,$$

Then

$$\|x(t) - y(t)\| \le \frac{\epsilon c}{\eta + cL}\left(e^{(\eta + cL)t} - 1\right), \quad \text{for all } t \ge 0$$

when

$$\|e^{At}\|_i \le ce^{\eta t}, \tag{8.3}$$

where $\|\cdot\|_i$ is the induced norm associated with the norm $\|\cdot\|$ and $c \ge 1$, $\eta + cL \ne 0$, $\eta \ne 0$, and $L > 0$.

Proof. (See Appendix 8.9.) □

Corollary 8.5 *In some special cases not covered above we have:*

1. *If $L = 0$ but $\eta \ne 0$, then*

$$\|x(t) - y(t)\| \le \frac{\epsilon c}{\eta}\left(e^{\eta t} - 1\right).$$

2. *If $\eta = 0$ and $L = 0$, then*

$$\|x(t) - y(t)\| \le c\epsilon t.$$

3. *If $\eta = 0$ but $L > 0$, then*

$$\|x(t) - y(t)\| \le \frac{\epsilon}{L}\left(e^{cLt} - 1\right).$$

4. *If $\eta \neq 0$ and $L > 0$ but $\eta + cL = 0$ (this means $\eta < 0$), then*

$$\|x(t) - y(t)\| \leq \frac{\epsilon c}{(-\eta)} \left[cLt + e^{-cLT} - e^{\eta t} \right].$$

Proof.   (See Appendix 8.9.) $\square$

The similarity of Lemmas 8.3 and 8.4 is easy to see. Their proofs are also similar. The most important distinction arises when A is stable and η can be chosen negative. Indeed, if $\eta + cL < 0$, then we can guarantee nondivergence of the solutions.

The proof can easily be extended to the case where A is time-varying:

Corollary 8.6 *Lemma 8.4 and Corollary 8.5 hold when A is time varying, with Equation (8.3) replaced by*

$$\|\Phi(t, s)\|_i \leq c e^{\eta(t-s)},$$

where $\Phi(t, s)$ is the transition matrix of the time-varying linear matrix $A(t)$.

Proof.   Proof is the same as that for Lemma 8.4, replacing $e^{A(t-s)}$ by $\Phi(t, s)$. $\square$

Then, the case $L = 0$ subsumes some of the global results of stability under persistent disturbances, e.g., [11, p. 167].

§8.4   SINGULAR PERTURBATIONS

The standard singular perturbation model is [87]

$$\begin{align}
\dot{x} &= f(x, z, \epsilon, t), & x(t_0) &= x_0, & x &\in \mathbb{R}^n, & \text{(8.4)}\\
\epsilon \dot{z} &= g(x, z, \epsilon, t), & z(t_0) &= z_0, & z &\in \mathbb{R}^m, & \text{(8.5)}
\end{align}$$

in which the derivatives of some of the states are multiplied by a small positive scalar ϵ. When we set $\epsilon = 0$, the state-space dimension reduces from $n + m$ to n and the second differential equation degenerates into an algebraic equation. Thus, Equation (8.4) represents a reduced-order model, with the resulting parameter perturbation being "singular." The reason for this terminology is seen when we divide both sides of Equation (8.5) by ϵ and let it approach zero.

We make use of the simpler model

$$\begin{align}
\dot{x} &= f(x, z, t), & x(t_0) &= x_0, & x &\in \mathbb{R}^n,\\
\dot{z} &= \alpha^2 [g(x, t) - z], & z(t_0) &= z_0, & z &\in \mathbb{R}^m,
\end{align}$$

where we have $\epsilon = 1/\alpha^2$, α a nonzero real number. With this rewriting, one sees why Equation (8.5) is said to represent the "fast transients," or fast dynamics. The following lemma shows explicitly a certain case where the dynamics can be made so fast that the resulting "tracking error" between $u(t) \equiv g(x(t), t)$ and $z(t)$ is kept small.

Lemma 8.7 *Let*

$$\dot{z}(t) = \alpha^2 (u(t) - z(t)),$$

where u is a Lipschitz continuous (with constant L) function of time. Given any $\epsilon > 0$, if

$$|z(0) - u(0)| = \epsilon_0 < \epsilon,$$

we can choose α large enough so that

$$|z(t) - u(t)| < \epsilon, \qquad t \geq 0.$$

Proof. (See Appendix 8.9.) □

The result can be extended to higher dimensions as follows:

Lemma 8.8 *Let*

$$\dot{z}(t) = \alpha^2 A(u(t) - z(t)),$$

where u and z are elements of $\mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. Further, assume that A is positive definite and that each coordinate of u is a Lipschitz continuous (with constant L) function of time. Given any $\epsilon > 0$, if

$$\|z(0) - u(0)\| = \epsilon_0 < \epsilon,$$

we can choose α large enough so that

$$\|z(t) - u(t)\| < \epsilon, \qquad t \geq 0.$$

Proof. Similar to the proof of Lemma 8.7. [Hint: consider the time derivative of $e^T e$, $e = (z - u)$, and use equivalence of norms on $\mathbb{R}^n$.] □

These lemmas allow us to use the robustness lemmas of the previous section to analyze certain singular perturbation problems. The idea of the preceding lemmas is that the fast dynamics are such that they maintain the corresponding algebraic equation, $z(t) = u(t)$, to within a small deviation (cf. invariant manifolds [87, p. 18]).

§8.5   MULTIPLE LYAPUNOV FUNCTIONS

In this section, we discuss Lyapunov stability of switched systems via **multiple Lyapunov functions (MLF)** The idea here is that even if we have Lyapunov functions for each system f_i individually, we need to impose restrictions on switching to guarantee stability. Indeed, it is easy to construct examples of two globally exponentially stable systems and a switching scheme that sends all trajectories to infinity:

Example 8.9 *Consider $f_1(x) = Ax$ and $f_2(x) = Bx$ where*

$$A = \begin{bmatrix} -0.1 & 1 \\ -10 & -0.1 \end{bmatrix}, \qquad B = \begin{bmatrix} -0.1 & 10 \\ -1 & -0.1 \end{bmatrix}.$$

Then $\dot{x} = f_i(x)$, is globally exponentially stable for $i = 1, 2$. But the switched system using f_1 in the second and fourth quadrants and f_2 in the first and third quadrants is unstable. See Figures 8-1–8-3, which plot ten seconds of trajectories for f_1, f_2, and the switched system starting from $(1, 0)$, $(0, 1)$, $(10^{-6}, 10^{-6})$, respectively.

We assume the reader is familiar with basic Lyapunov theory (continuous and discrete time), say, at the level of [96]. The level of rigor of the proofs is similar to those in that book. We let $S(r)$, $B(r)$, and $\overline{B}(r)$ represent the sphere, ball, and closed ball of Euclidean radius r about the origin in $\mathbb{R}^n$, respectively.

Below, we deal with systems that switch among vector fields (resp. difference equations), over time or regions of state-space. One can associate with such a system the following
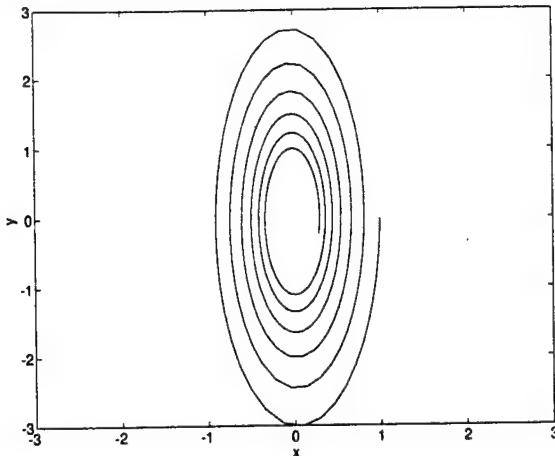
Figure 8-1: Trajectory of f_1.
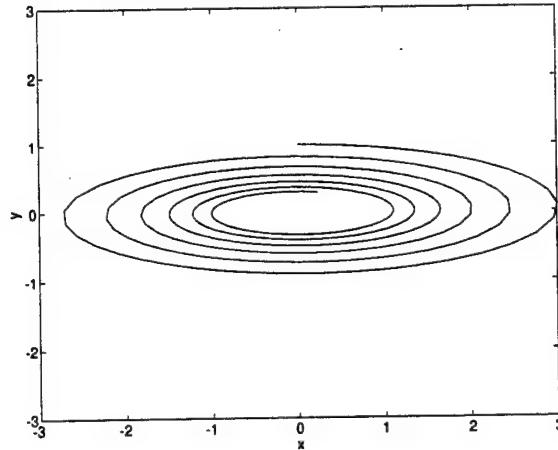
Figure 8-2: Trajectory of f_2.

(anchored) **switching sequence**, indexed by an initial state, x_0:

$$S = x_0; \ (i_0, t_0), (i_1, t_1), \ldots, (i_N, t_N), \ldots. \tag{8.6}$$

The sequence may or may not be infinite. In the finite case, we may take $t_{N+1} = \infty$, with all further definitions and results holding. However, we present in the sequel only in the infinite case to ease notation. The switching sequence, along with Equation (4.6), completely describes the trajectory of the system according to the following rule: (i_k, t_k) means that the system evolves according to $\dot{x}(t) = f_{i_k}(x(t), t)$ for $t_k \le t < t_{k+1}$. We denote this trajectory by $x_S(\cdot)$. Throughout, we assume that the switching sequence is **minimal** in the sense that $i_j \ne i_{j+1}$, $j \in \mathbb{Z}_+$.

We can take projections of this sequence onto its first and second coordinates, yielding the sequence of indices,

$$\pi_1(S) = x_0; \ i_0, i_1, \ldots, i_N, \ldots,$$
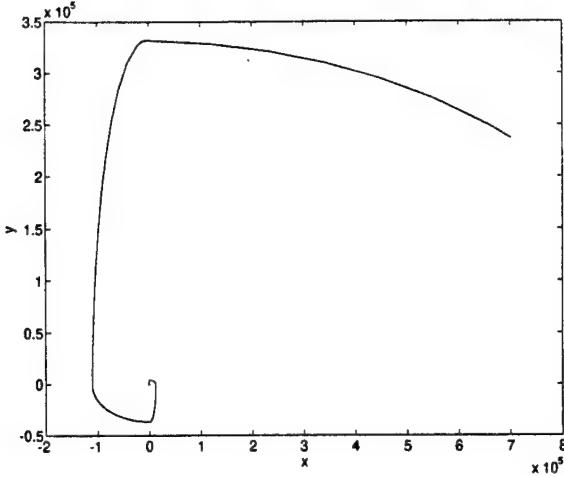
Figure 8-3: Trajectory of switched system.

and the sequence of switching times,

$$\pi_2(S) = x_0; \ t_0, t_1, \ldots, t_N, \ldots,$$

respectively. Suppose S is a switching sequence as in Equation (8.6). We denote by $S|i$ the sequence of switching times whose corresponding index is i for the discrete case and the endpoints of the times that system i is active in the continuous-time case. The **interval completion** $\mathcal{I}(T)$ of a strictly increasing sequence of times $T = t_0, t_1, \ldots, t_N, \ldots$, is the set

$$\bigcup_{j \in \mathbb{Z}_+} (t_{2j}, t_{2j+1}).$$

Finally, let $\mathcal{E}(T)$ denote the **even sequence** of T:

$$t_0, t_2, t_4, \ldots .$$

Below, we say that V is a **candidate Lyapunov function** if V is a continuous, positive definite function (about the origin, 0) with continuous partial derivatives. Note this assumes $V(0) = 0$. We also use

Definition 8.10 *Given a strictly increasing sequence of times T in $\mathbb{R}$ (resp. $\mathbb{Z}$), we say that V is* **Lyapunov-like** *for function f and trajectory $x(\cdot)$ (resp. $x[\cdot]$) over T if*

- $\dot{V}(x(t)) \leq 0$ *(resp. $V(x[t+1]) \leq V(x[t])$) for all $t \in \mathcal{I}(t)$ (resp. $t \in T$),*

- V *is monotonically nonincreasing on $\mathcal{E}(T)$ (resp. T).*

Theorem 8.11 *Suppose we have candidate Lyapunov functions V_i, $i = 1, \ldots, N$, and vector fields $\dot{x} = f_i(x)$ (resp. difference equations $x[k+1] = f_i(x[k])$) with $f_i(0) = 0$, for all i. Let $\mathcal{S}$ be the set of all switching sequences associated with the system.*

If for each $S \in \mathcal{S}$ we have that for all i, V_i is Lyapunov-like for f_i and $x_S(\cdot)$ over $S|i$, then the system is stable in the sense of Lyapunov.

Proof. In each case, we do the proofs only for $N = 2$.

- **Continuous-time:** Let $R > 0$ be arbitrary. Let $m_i(\alpha)$ denote the minimum value of V_i on $S(\alpha)$. Pick $r_i < R$ such that in $B(r_i)$ we have $V_i < m_i(R)$. This choice is possible via the continuity of V_i. Let $r = \min(r_i)$. With this choice, if we start in $B(r)$, either vector field alone will stay within $B(R)$.

  Now, pick $\rho_i < r$ such that in $B(\rho_i)$ we have $V_i < m_i(r)$. Set $\rho = \min(\rho_i)$. Thus, if we start in $B(\rho)$, either vector field alone will stay in $B(r)$. Therefore, whenever the other is first switched on we have $V_i(x(t_1)) < m_i(R)$, so that we will stay within $B(R)$.

- **Discrete-time:** Let $R > 0$ be arbitrary. Let $m_i(\alpha, \beta)$ denote the minimum value of V_i on the closed annulus $\overline{B}(\beta) - B(\alpha)$. Pick $R_0 < R$ so that none of the f_i can jump out of $B(R)$ in one step. Pick $r_i < R_0$ such that in $B(r_i)$ we have $V_i < m_i(R_0, R)$. This choice is possible via the continuity of V_i. Let $r = \min(r_i)$. With this choice, if we start in $B(r)$, either equation alone will stay within $B(R)$.

  Pick $r_0 < r$ so that none of the f_i can jump out of $B(r)$ in one step. Now, pick $\rho_i < r_0$ such that in $B(\rho_i)$ we have $V_i < m_i(r_0, r)$. Set $\rho = \min(\rho_i)$. Thus, if we start in $B(\rho)$, either equation alone will stay in $B(r_0)$, and hence $B(r)$. Therefore, whenever the other is first switched on we have $V_i(x(t_1)) < m_i(R_0, R)$, so that we will stay within $B(R_0)$, and hence $B(R)$.

The proofs for general N require N sets of concentric circles constructed as the two were in each case above. $\qquad\square$

Some remarks are in order:

- The case $N = 1$ is the usual theorem for Lyapunov stability [96]. Also, compare Figures 8-4 and 8-5, both of which depict the continuous-time case.

- The theorem also holds if the f_i are time-varying.

- It is easy to see that the theorem does not hold if $N = \infty$, and we leave it to the reader to construct examples.
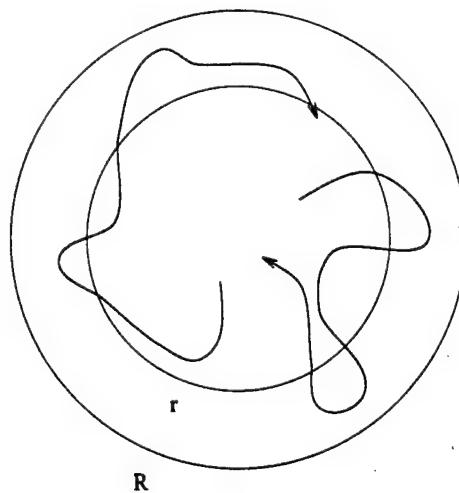

Figure 8-4: Lyapunov stability.
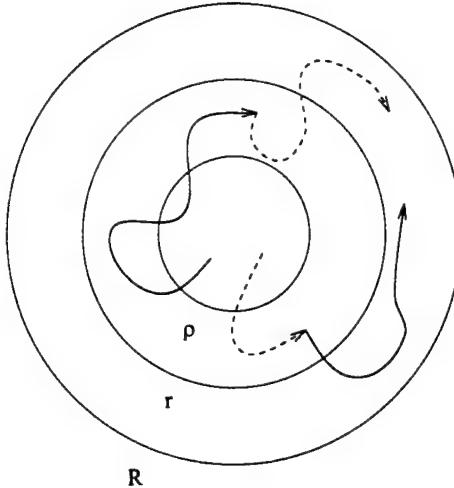
Figure 8-5: Multiple Lyapunov stability, $N = 2$.

Example 8.12 *Pick any line through the origin. Going back to Example 8.9 and choosing to use f_1 above the line and f_2 below it, the resulting system is globally asymptotically stable. The reason is that each system is strictly stable linear and hence diminishes $V_i = x^T P_i x$ for some $P_i > 0$. However, since switchings occur on a line through the origin, we are assured that on switches to system i, V_i is lower energy than when it was last switched out.*

It is possible to use different conditions on the V_i to ensure stability. For instance, consider the following

Definition 8.13 *If there are candidate Lyapunov functions V_i corresponding to f_i for all i, we say they satisfy the* **sequence nonincreasing condition** *for a trajectory $x(\cdot)$ if*

$$V_{i_{j+1}}(x(t_{j+1})) < V_{i_j}(x(t_j)).$$

This is a stronger notion than the Lyapunov-like condition used above.

The sequence nonincreasing condition is used in the stability (version of the asymptotic stability) theorem of [119]. Thus that theorem is a special case of the continuous-time version of Theorem 8.11 above. Moreover, the proof of asymptotic stability in [119] is flawed since it only proves state convergence and not state convergence plus stability, as required. It can be fixed using our theorem.

Now, consider the case where the index set is an arbitrary compact set:

$$\dot{x} = f(x, \lambda), \qquad \lambda \in K, \text{ compact.} \tag{8.7}$$

Here, $x \in \mathbb{R}^n$ and f is globally Lipschitz in x, continuous in λ. For brevity, we only consider the continuous-time case. Again, we assume finite switches in finite time.

As above, we may define a switching sequence

$$S = x_0; \ (\lambda_0, t_0), (\lambda_1, t_1), \ldots, (\lambda_N, t_N), \ldots.$$

with its associated projection sequences.

Theorem 8.14 *Suppose we have candidate Lyapunov functions $V_\lambda \equiv V(\cdot, \lambda)$ and vector fields as in Equation (8.7) with $f(0, \lambda) = 0$, for each $\lambda \in K$. Also, $V : \mathbb{R}^n \times K \to \mathbb{R}_+$ is*

continuous. Let $\mathcal{S}$ be the set of all switching sequences associated with the system.

If for each $S \in \mathcal{S}$ we have that for all i, V_λ is Lyapunov-like function for f_λ and $x_S(\cdot)$ over $S|\lambda$, and the V_λ satisfy the sequence nonincreasing condition for $x_S(\cdot)$, then the system is stable in the sense of Lyapunov.

Proof. We present the proof in the case that K is sequentially compact, which is automatic if K is a metric space. The general case follows with little change from the argument below by using countable compactness and nets instead of sequences. (See [62, 113] for definitions).

The Lyapunov-like and sequence nonincreasing constraints are such that if $\pi_i(S) = x_0;\ \lambda_0, \lambda_1, \lambda_2, \ldots$, then the state $x(t)$ will remain within the set:

$$R_{V(x_0,\lambda_0)} \equiv \bigcup_{\lambda \in K} \{x \mid V(x,\lambda) < V(x_0,\lambda_0)\}.$$

Next, note that if x_0 lies in

$$I_\epsilon \equiv \left\{ x \mid \sup_{\lambda \in K} V(x,\lambda) < \epsilon \right\},$$

then the state will remain in R_ϵ.

Thus, it remains to show that given any $\epsilon > 0$, there exist $\epsilon', \delta > 0$ such that

$$B(\delta) \subset I_{\epsilon'} \cap B(\epsilon) \subset R_{\epsilon'} \cap B(\epsilon) \subset B(\epsilon).$$

Letting m denote the minimum of V on $S(\epsilon) \times K$, $\epsilon' = m/2$ satisfies the last equation. Now $I_{\epsilon'}$ contains the origin, $0 \in \mathbb{R}^n$. Suppose there is no open ball about 0 in $I_{\epsilon'}$. Then for each $n \in \mathbb{Z}_+$, there exists y_n such that

$$\|y_n\| \leq 1/n, \qquad \sup_{\lambda \in K} V(y_n,\lambda) \geq \epsilon'.$$

Further, we may take each of the y_n distinct. Let $\lambda_n \in K$ be the point at which the sup above is attained. Since $\overline{B}(\epsilon') \times K$ is sequentially compact, there is a subsequence $\{(y_{i_k}, \lambda_{i_k})\}$ converging to $(0, \lambda^*)$ with $V(y_{i_k}, \lambda_{i_k}) \geq \epsilon'$, a contradiction to the continuity of V and the assumption that $V(0,\lambda) = 0$ for all $\lambda \in K$. $\qquad\square$

This theorem is a different generalization of the aforementioned theorem in [119].

§8.6 ITERATED FUNCTION SYSTEMS

In this section, we study iterated function systems theory as a tool for Lagrange stability. We begin with some background from [12, 143, 48]:

Definition 8.15 *Recall that a **contractive function** f is one such that there exists $s < 1$ where $d(f(x), f(y)) \leq sd(x,y)$, for all x, y.*

*An **iterated function system** or **IFS** is a complete metric space and a set $\{f_i\}_{i \in I}$ of contractive functions such that I is a compact space and the map $(x,i) \mapsto f_i(x)$ is continuous.*

The image of a compact set X under an IFS is the set $Y = \bigcup_{i \in I} f_i(X)$. It is compact. Now suppose W is an IFS. Let $S(W)$ be the semigroup generated by W under composition.

For example, if $W = \{f, g\}$ then

$$S(W) = f, g, f \circ f, f \circ g, g \circ f, g \circ g, \ldots .$$

Now, define A_W to be the closure of the fixed points of $S(W)$. We have

Theorem 8.16 *Suppose $W = \{w_i\}_{i \in I}$ is an IFS on X. Then*

- *A_W is compact.*

- *$A_W = \bigcup_{i \in I} w_i(A_W)$.*

- *For all $x \in X$,*

$$A_W = \bigcup_\sigma \left\{ \lim_{n \to \infty} w_{\sigma_1} \circ w_{\sigma_2} \circ \cdots \circ w_{\sigma_n}(x) \right\},$$

  where $\sigma = (\sigma_1, \sigma_2, \ldots)$, $\sigma_i \in I$.

The relevance of this theorem is twofold:

- A_W is an invariant set under the maps $\{w_i\}_{i \in I}$.

- All points approach A_W under iterated composition of the maps $\{w_i\}_{i \in I}$.

Clearly, this theory can be applied in the case of a set of contractive discrete maps indexed by a compact set (usually finite). Thus, it is directly applicable to systems of the form Equation (4.7).

Example 8.17 *The following IFS is well-known:*

$$F_i(x) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} x + \begin{bmatrix} 0.5 \cdot 1_{\{i>1\}} \\ 0.5 \cdot 1_{\{i<3\}} \end{bmatrix}, \qquad i = 1, 2, 3.$$

It is pictured in Figure 8-6.

To obtain contractive maps while switching among differential equations requires a little thought. Assume there is some lower limit T on the inter-switching time. Now, notice that for any inter-switching time $r \geq T$, there is a decomposition into smaller intervals as follows:

$$r = \sum_{i=1}^M t_i, \qquad t_i \in [T, 2T].$$

Proof. Let $k = \lfloor r/(2T) \rfloor$ and $q = r - 2Tk$. Now, $2T > q \geq 0$. If $q = 0$, the decomposition is $t_i = 2T$, $i = 1, \ldots, k$. If $2T > q \geq T$, the decomposition is $t_i = 2T$, $i = 1, \ldots, k$; $t_{k+1} = q$; the first equation not applying if $k = 0$. Finally, if $T > q > 0$, then (we must have $k \geq 1$ since $r \geq T$) and $2T > q + T > T$, so the decomposition is $t_i = 2T$, $i = 1, \ldots, k-1$; $t_k = T$; $t_{k+1} = T + q$; the first equation not applying if $k = 1$. $\square$

Therefore, we can convert switching among vector fields into an IFS by letting $I = \bigcup_{j=1,\ldots,N} j \times [T, 2T]$. In particular, we see that for each i, if it is active for a time $r \geq T$, we can write the solution in that interval as $\phi_r^i(x) = (\circ_{j=1}^M \phi_{t_j}^i)(x)$, where ϕ_t^i is the fundamental solution for f_i acting for time t. Thus the switching sequence can be converted to an iterated composition of maps indexed by the compact set I.

The other interesting point about IFS theory is that the different vector fields (or difference equations) need not have the same equilibrium point. This is important as it appears to be the usual case in switched and hybrid systems (cf. Example 3.4).
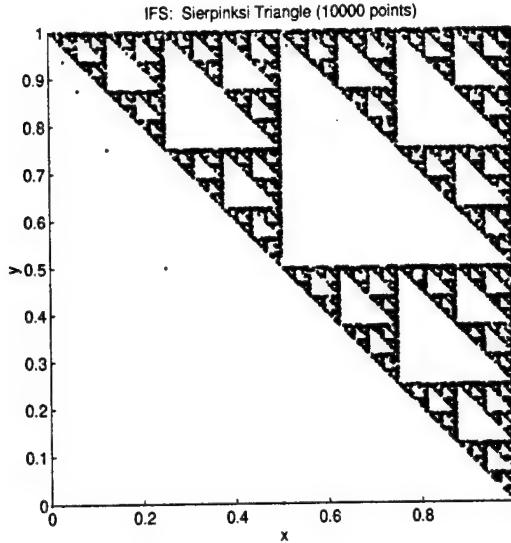
Figure 8-6: Example IFS.

Example 8.18 *Starting with Example 8.17, we consider a corresponding* **differential IFS** **(DIFS)**, *with*

$$F_i(x) = \begin{bmatrix} -\alpha & 0 \\ 0 & -\alpha \end{bmatrix} x + \begin{bmatrix} \alpha \cdot 1_{\{i>1\}} \\ \alpha \cdot 1_{\{i<3\}} \end{bmatrix}, \qquad i = 1, 2, 3.$$

It is pictured in Figure 8-7 for $\alpha = \ln 2$ *and* $T = 2/3$.

In conclusion, in IFS we have a tool for analyzing the Lagrange stability and computing the invariant sets of switched systems of the form Equations (4.6) and (4.7). The resulting sets A_W are reminiscent of those for usual IFS (see [12]), although we don't give any here. The reader may consult [12] for algorithms to compute such invariant sets.

§8.7 DISCUSSION

In both the MLF and IFS cases, the stability results are sufficiency conditions on the continuous dynamics and switching. This work represents the rudiments of a stability theory of the systems in Equations (4.6) and (4.7) and, in turn, of hybrid systems. We also discussed the case where $\{1, \ldots, N\}$ in Equations (4.6) and (4.7) is replaced by an arbitrary compact set.

For future directions, we offer the following brief treatment. In searching for necessary and sufficient stability criteria, we expect that the theory in [94] appears helpful. An early use of our MLF theory is given in [147], which deals with convergence of a combined scheme for robotic planning and obstacle avoidance. As far as IFS, we have yet to explore their full potential. For instance, we can state IFS theorems analogous to Theorem 8.11, namely, in which the maps need only be contractions on the points (time periods) on which they are applied. Finally, if there is no lower limit T on the inter-switching time, then we are not assured to have a contraction mapping. However, as long as we have only finite switches in
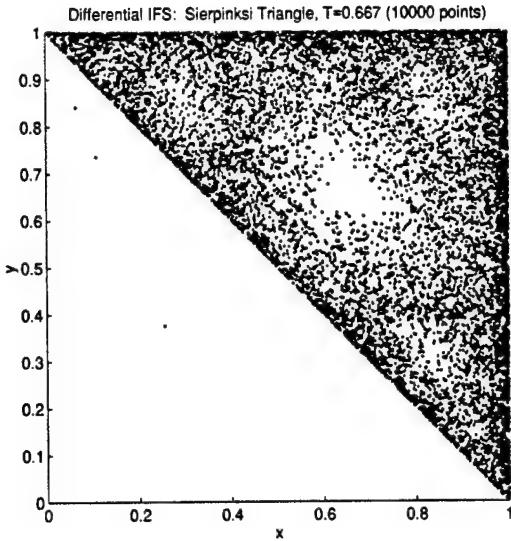
Figure 8-7: Example DIFS.

finite time, one expects that the trajectories should be well-behaved (e.g., invoke continuity of ODE solutions and take convex hulls).

Finally, it is not hard to generalize our MLF theory to the case of different equilibria, which is generally the case in hybrid systems. For example, under a Lyapunov-like switching rule, after all controllers have been switched in at level α_i, the set $\bigcup_i V_i^{-1}(\alpha_i)$ is invariant.

§8.8 NOTES

The work in §§8.2–8.4 first appeared in [21], later summarized in [28]. The work in §8.5 and §8.6 was begun in [23] and continued in [30]. The extension of Bendixson's Theorem and the Linear Robustness Lemma have applicability beyond the systems discussed in this thesis. The Linear Robustness Lemma was published in [29].

After this work was published, we became aware of the related work in [118]. There Pavlidis concludes stability of differential equations containing impulses by introducing a positive definite function which decreases during the occurrence of an impulse and remains constant or decreases during the "free motion" of the system. Hence, it is a special case of our results.

In personal discussions, Prof. Wyatt S. Newman essentially conjectured Theorem 8.11 in the continuous-time setting.

§8.9 APPENDIX A: ASSORTED PROOFS

§8.9.1 CONTINUITY LEMMAS

The proofs of our continuity lemmas depend critically on the well-known Bellman-Gronwall inequality [58, p. 252]:

Lemma 8.19 (Bellman-Gronwall) *Let*

1. f, g, k; $\mathbb{R}_+ \to \mathbb{R}$ and locally integrable;

2. $g \geq 0$, $k \geq 0$;

3. $g \in L_e^\infty$;

4. gk is locally integrable on $\mathbb{R}_+$.

Under these conditions, if $u : \mathbb{R}_+ \to \mathbb{R}$ satisfies

$$u(t) \leq f(t) + g(t) \int_0^t k(\tau)u(\tau)d\tau, \qquad \text{for all } t \in \mathbb{R}_+,$$

then

$$u(t) \leq f(t) + g(t) \int_0^t k(\tau)f(\tau) \left[\exp \int_\tau^t k(\sigma)g(\sigma)d\sigma\right] d\tau, \qquad \text{for all } t \in \mathbb{R}_+.$$

Proof. [of Lemma 8.3] For any $t \geq 0$,

$$x(t) = x_0 + \int_0^t F(x, \tau)d\tau,$$

$$y(t) = x_0 + \int_0^t G(y, \tau)d\tau.$$

Subtracting yields

$$
\begin{aligned}
x(t) - y(t) &= \int_0^t F(x, \tau)d\tau - \int_0^t G(y, \tau)d\tau \\
&= \int_0^t [F(x, \tau) - F(y, \tau) + F(y, \tau) - G(y, \tau)]d\tau, \\
\|x(t) - y(t)\| &= \left\| \int_0^t [F(x, \tau) - F(y, \tau) + F(y, \tau) - G(y, \tau)]d\tau \right\| \\
&\leq \int_0^t \|F(x, \tau) - F(y, \tau)\|d\tau + \int_0^t \|F(y, \tau) - G(y, \tau)\|d\tau \\
&\leq L \int_0^t \|x(\tau) - y(\tau)\|d\tau + \epsilon t.
\end{aligned}
$$

Using the Bellman-Gronwall Lemma, we obtain

$$
\begin{aligned}
\|x(t) - y(t)\| &\leq \epsilon t + \int_0^t L\epsilon\tau \left[\exp \int_\tau^t Ld\sigma\right] d\tau \\
&= \epsilon\left\{t + L\int_0^t \tau e^{L(t-\tau)}d\tau\right\} \\
&= \epsilon\left\{t + Le^{Lt}\int_0^t \tau e^{-L\tau}d\tau\right\}.
\end{aligned}
$$

If $L = 0$

$$\|x(t) - y(t)\| = \epsilon t,$$

and if $L > 0$, we compute

$$\int_0^t \tau e^{-L\tau} d\tau = \left[\frac{e^{-L\tau}}{L^2} (-L\tau - 1) \right]_0^t$$

$$= -\frac{e^{-Lt}}{L^2}(Lt + 1) + \frac{1}{L^2}.$$

Therefore,

$$\|x(t) - y(t)\| \leq \epsilon \left\{ t - t - \frac{1}{L} + \frac{1}{L} e^{Lt} \right\}$$

$$= \frac{\epsilon}{L} \left(e^{Lt} - 1 \right).$$

$\square$

Proof. [of Lemma 8.4] For any $t \geq 0$,

$$x(t) = e^{At} x_0 + \int_0^t e^{A(t-\tau)} F(x, \tau) d\tau,$$

$$y(t) = e^{At} x_0 + \int_0^t e^{A(t-\tau)} G(y, \tau) d\tau.$$

Subtracting yields

$$x(t) - y(t) = \int_0^t e^{A(t-\tau)} [F(x, \tau) - F(y, \tau) + F(y, \tau) - G(y, \tau)] d\tau,$$

$$\|x(t) - y(t)\| \leq \int_0^t \|e^{A(t-\tau)}\|_i \|F(x, \tau) - F(y, \tau)\| d\tau + \int_0^t \|e^{A(t-\tau)}\|_i \|F(y, \tau) - G(y, \tau)\| d\tau$$

$$\leq L \int_0^t \|e^{A(t-\tau)}\|_i \|x(\tau) - y(\tau)\| d\tau + \epsilon \int_0^t \|e^{A(t-\tau)}\|_i d\tau.$$

Now we are given that $\|e^{As}\|_i \leq c e^{\eta s}$, so that

$$\int_0^t \|e^{A(t-\tau)}\|_i d\tau \leq c e^{\eta t} \int_0^t e^{-\eta \tau} d\tau$$

$$\leq \frac{c}{\eta} e^{\eta t} \left(1 - e^{-\eta t} \right)$$

$$\leq \frac{c}{\eta} \left(e^{\eta t} - 1 \right). \tag{8.8}$$

Therefore,

$$\|x(t) - y(t)\| \leq c L e^{\eta t} \int_0^t e^{-\eta \tau} \|x(\tau) - y(\tau)\| d\tau + \frac{\epsilon c}{\eta} \left(e^{\eta t} - 1 \right). \tag{8.9}$$

Using the Bellman-Gronwall Lemma, we obtain

$$\|x(t) - y(t)\| \leq \frac{\epsilon c}{\eta} \left(e^{\eta t} - 1 \right) + c L e^{\eta t} \int_0^t e^{-\eta \tau} \frac{\epsilon c}{\eta} (e^{\eta \tau} - 1) \left[\exp \int_\tau^t e^{-\eta \sigma} c L e^{\eta \sigma} d\sigma \right] d\tau$$

$$= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + c L e^{\eta t} \int_0^t \left(1 - e^{-\eta \tau} \right) e^{cL(t-\tau)} d\tau \right]$$

$$
= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + cLe^{\eta t}e^{cLt} \int_0^t \left(e^{-cL\tau} - e^{-(\eta+cL)\tau} \right) d\tau \right] \tag{8.10}
$$

$$
= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + cLe^{(\eta+cL)t} \left(\frac{1}{cL} - \frac{1}{cL}e^{-cLt} + \frac{1}{\eta+cL}e^{-(\eta+cL)t} - \frac{1}{\eta+cL} \right) \right]
$$

$$
= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + e^{(\eta+cL)t} - e^{\eta t} + \frac{cL}{\eta+cL} - \frac{cL}{\eta+cL}e^{(\eta+cL)t} \right]
$$

$$
= \frac{\epsilon c}{\eta} \left[\frac{cL}{\eta+cL} - 1 \right] \left[1 - e^{(\eta+cL)t} \right]
$$

$$
= \frac{\epsilon c}{\eta} \left[\frac{-\eta}{\eta+cL} \right] \left[1 - e^{(\eta+cL)t} \right]
$$

$$
= \frac{\epsilon c}{\eta+cL} \left[e^{(\eta+cL)t} - 1 \right].
$$

$\square$

Proof. [of Corollary 8.5] Now we deal some special cases not covered above:

1. If $L = 0$ but $\eta \neq 0$, then Equation (8.9) gives

$$
\|x(t) - y(t)\| \leq \frac{\epsilon c}{\eta} \left(e^{\eta t} - 1 \right).
$$

If $\eta = 0$ then Equation (8.8) is replaced by

$$
\int_0^t \|e^{A(t-\tau)}\|_i d\tau \leq ct. \tag{8.11}
$$

2. So, if $\eta = 0$ and $L = 0$

$$
\|x(t) - y(t)\| \leq c\epsilon t.
$$

3. If $\eta = 0$ and $L > 0$ then Equation (8.9) is replaced by

$$
\|x(t) - y(t)\| \leq cL \int_0^t \|x(\tau) - y(\tau)\| d\tau + \epsilon ct.
$$

in which case the Bellman-Gronwall Lemma gives

$$
\begin{aligned}
\|x(t) - y(t)\| &\leq \epsilon ct + cL \int_0^t \epsilon c\tau \left[\exp \int_\tau^t cL \, d\sigma \right] d\tau \\
&\leq \epsilon ct + \epsilon c^2 L \int_0^t \tau e^{cL(t-\tau)} d\tau \\
&\leq \epsilon ct + \epsilon c^2 L e^{cLt} \int_0^t \tau e^{cL\tau} d\tau.
\end{aligned}
$$

Now repeating the calculation of Equation (8.8) with cL identified with L:

$$
\begin{aligned}
\|x(t) - y(t)\| &\leq \epsilon ct + \epsilon c^2 L e^{cLt} \left[-\frac{e^{-cLt}}{(cL)^2}(Lt + 1) + \frac{1}{(cL)^2} \right] \\
&\leq \epsilon ct - \epsilon ct - \frac{\epsilon}{L} + \frac{\epsilon}{L}e^{cLt} \\
&\leq \frac{\epsilon}{L} \left(e^{cLt} - 1 \right).
\end{aligned}
$$

4. If $\eta \neq 0$ and $L > 0$ but $\eta + cL = 0$ (this means $\eta < 0$), then Equation (8.10) and the further computations simplify to

$$
\begin{aligned}
\|x(t) - y(t)\| &= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + cL \int_0^t \left(e^{-cL\tau} - 1 \right) d\tau \right] \\
&= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + cL \left(\frac{1}{cL} - \frac{1}{cL} e^{-cLt} - t \right) \right] \\
&= \frac{\epsilon c}{\eta} \left[e^{\eta t} - 1 + 1 - e^{-cLt} - cLt \right] \\
&= \frac{\epsilon c}{\eta} \left[e^{\eta t} - e^{-cLt} - cLt \right].
\end{aligned}
$$

$\square$

§8.9.2 SINGULAR PERTURBATION LEMMAS

Proof. [of Lemma 8.7] Let $e = z - u$. Then

$$
\begin{aligned}
\frac{d^+ e}{dt} &= \frac{d^+ z}{dt} - \frac{d^+ u}{dt} \\
&= \alpha^2 (u - z) - \frac{d^+ u}{dt} \\
&= -\alpha^2 e - \frac{d^+ u}{dt},
\end{aligned}
$$

where

$$
\frac{d^+ z}{dt} = \lim_{h \to 0^+} \frac{z(t + h) - z(t)}{h}.
$$

Now, since u is Lipschitz, we have

$$
\left| \frac{d^+ u}{dt} \right| \leq L.
$$

Thus, if we choose α such that $\alpha^2 > L/\epsilon$, then when $e \geq \epsilon$, we have

$$
\frac{d^+ e}{dt} \leq -\alpha^2 \epsilon + L < 0.
$$

Similarly, when $e \leq -\epsilon$, we have

$$
\frac{d^+ e}{dt} \geq \alpha^2 \epsilon - L > 0.
$$

Thus, the set $|e| < \epsilon$ is an invariant set. $\square$

§8.10 APPENDIX B: BENDIXSON EXTENSION

This appendix treats the background, statement, and proof of our extension of Bendixson's Theorem.

Bendixson's theorem gives conditions under which a region cannot contain a periodic solution (or limit cycle). It is usually stated as follows (statement and footnote adapted from [142, pp. 31-32]):

Theorem 8.20 (Bendixson's Theorem) *Suppose D is a simply connected[2] domain in $\mathbb{R}^2$ such that the quantity $\nabla f(x)$ (the divergence of f) defined by*

$$\nabla f(x) = \frac{\partial f_1}{\partial x_1}(x_1, x_2) + \frac{\partial f_2}{\partial x_2}(x_1, x_2)$$

is not identically zero over any subregion of D and does not change sign in D. Then D contains no closed trajectories of

$$\dot{x}_1(t) = f_1[x_1(t), x_2(t)],$$
$$\dot{x}_2(t) = f_2[x_1(t), x_2(t)].$$

The proof of Bendixson's Theorem depends, in a critical way, on Green's Theorem. The usual statement of Green's Theorem says [109] that a C^1 vector field $f(x)$ on a compact region A in $\mathbb{R}^n$ with C^1 boundary B satisfies

$$\int_B f(x) \cdot \mathbf{n}(A, x) d\sigma = \int_A \nabla f(x) d\mathcal{L}^n x,$$

where $\mathbf{n}(A, x)$ is the exterior unit normal to A at x, $d\sigma$ is the element of area on B, and $\mathcal{L}^n$ is the Lebesgue measure on $\mathbb{R}^n$. It is possible, however, to treat more general regions and vector fields that are merely Lipschitz continuous. A general extension is the so-called Gauss-Green-Federer Theorem given in [109]. Even the statement of this theorem requires the development of a bit of the language of geometric measure theory. We state a relaxed version of this theorem that is still suitable for our purposes. In the final formula, ∇f exists almost everywhere because a Lipschitz continuous function is differentiable almost everywhere.

Theorem 8.21 (Relaxation of Gauss-Green-Federer) *Let A be a compact region of $\mathbb{R}^n$ with C^1 boundary B. Then for any Lipschitz vector field $f(x)$,*

$$\int_B f(x) \cdot \mathbf{n}(A, x) d\sigma = \int_A \nabla f(x) d\mathcal{L}^n x.$$

Now we can prove our version of Bendixson's Theorem:

Proof. [of Theorem 8.1] The proof is similar to that of Bendixson's Theorem in [142, pp. 31–32].

Suppose, for contradiction, that J is a closed trajectory of Equations (8.1) and (8.2). Then at each point $x \in J$, the vector field $f(x)$ is tangent to J. Then $f(x) \cdot \mathbf{n}(S, x) = 0$ for all $x \in J$, where S is the area enclosed by J. But by Theorem 8.21

$$0 = \int_J f(x) \cdot \mathbf{n}(A, x) dl = \int_S \nabla f(x) d\mathcal{L}^2 x.$$

Therefore, we must have either (i) $\nabla f(x)$ is zero almost everywhere, or (ii) the sets $\{x \in S | \nabla f(x) < 0\}$ and $\{x \in S | \nabla f(x) > 0\}$ both have positive measure. But if S is a subset of D, neither can happen. Hence, D contains no closed trajectories of Equations (8.1)–(8.2). $\square$

---

[2]**A connected** region can be thought of as a set that is in one piece, i.e., one in which every two points in the set can be connected by a curve lying entirely within the set. A set is **simply connected** if (1) it is connected and (2) its boundary is connected.

Chapter 9

Analyzing Examples

In this chapter, the attention focuses on example systems and their analysis. We first analyze a class of two-state, continuous switched systems, proving global stability. Using tools from §8, we conclude stability of a class of continuations of those systems. The example systems arise from a realistic aircraft control problem.

§9.1 INTRODUCTION

In this chapter, we use the tools of §8 to analyze some example continuous switched systems motivated by a realistic aircraft control problem. In the next section, we present an example continuous switched control problem. This system is inspired from one used in the longitudinal control of modern aircraft such as the F-8 [134]. The control law uses a "logical" function (max) to pick between one of two stable controllers: the first a servo that tracks pilot inputs, the second a regulator about a fixed angle of attack. The desired effect of the total controller is to "track pilot inputs except when those inputs would cause the aircraft to exceed a maximum angle of attack." We analyze the stability of this hybrid system in the case where the pilot input is zero and the controllers are linear full-state feedback. We call this the max system. While the restriction to this case seems strong, one should note that the stability of such systems is typically verified only by extensive simulation [134]. In this chapter, we use the tools discussed above to *prove* nontrivial statements about the controller's behavior. For example, we show that no limit cycles exist by applying our extension of Bendixson's Theorem. We also show that the family of linear full-state feedback max systems can be reduced to a simpler family via a change of basis and analysis of equilibria. Finally, we give a Lyapunov function that proves that all systems of this canonical form are globally asymptotically stable. The Lyapunov function itself has a logical component, and the proof that it diminishes along trajectories is split into logical cases.

In §9.3 we analyze a "continuation" of the max system. Specifically, we use a dynamic variable (output of a differential equation) instead of the output given by the max function directly. This corresponds to a dynamical smoothing or switching hysteresis, as we motivated above. By using our lemma on the robustness of linear ODEs, we conclude stability properties of this (singular perturbation) continuation from those of the original max system.

The Appendix collects the more tedious proofs.

§9.2 EXAMPLE 1: MAX SYSTEM

As an example of a switched system, we consider a problem combining logic in a continuous control system. Specifically, we start with the system

$$\Sigma: \qquad \frac{d}{dt}\begin{bmatrix} q \\ \alpha \end{bmatrix} = \begin{bmatrix} -1 & -10 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} -1 \\ 0.1 \end{bmatrix}\delta,$$

$$\begin{bmatrix} \alpha \\ n_z \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -300 \end{bmatrix}\begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 30 \end{bmatrix}\delta,$$

or, symbolically,

$$\begin{aligned} \dot{x} &= Ax + B\delta, \\ \alpha &= C_1 x + D_1 \delta = C_1 x, \\ n_z &= C_2 x + D_2 \delta. \end{aligned}$$

These equations arise from the longitudinal dynamics of an aircraft (see Figure 9-1) with reasonable values chosen for the physical parameters. The variable θ is the pitch angle and α is the angle of attack. The input command δ is the angle of the elevator. The normal acceleration, n_z, is the output variable which we would like to track, i.e., we assume that the pilot requests desired values of n_z with his control stick. As a constraint, the output variable α must have a value not much larger than α_{lim} (for the model to be valid and the plane to be controlled adequately). A successful controller would satisfy both of these objectives simultaneously to the extent that this is possible: we desire good tracking of the pilot's input without violating the constraint $\alpha \le \alpha_{\text{lim}} + \epsilon$, for $\epsilon \ge 0$ some safety margin.
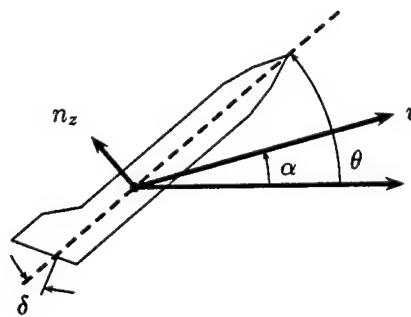
Figure 9-1: Longitudinal Aircraft View.

Now, suppose that two controllers, K_1 and K_2, have been designed to output δ_1 and δ_2 such that (1) Σ is regulated about $\alpha = \alpha_{\text{lim}}$ when $\delta = \delta_1$; and (2) Σ tracks command r—the pilot's desired n_z—when $\delta = \delta_2$, respectively. Finally, suppose that we add the following logical block: $\delta = \max(\delta_1, \delta_2)$. Control configurations much like this (see Figure 9-2) have been used to satisfy the objectives of our aircraft control problem [134].

To our knowledge, the stability of such systems has only been probed via extensive simulation [134]. In the remainder of this section, we examine the stability of certain cases
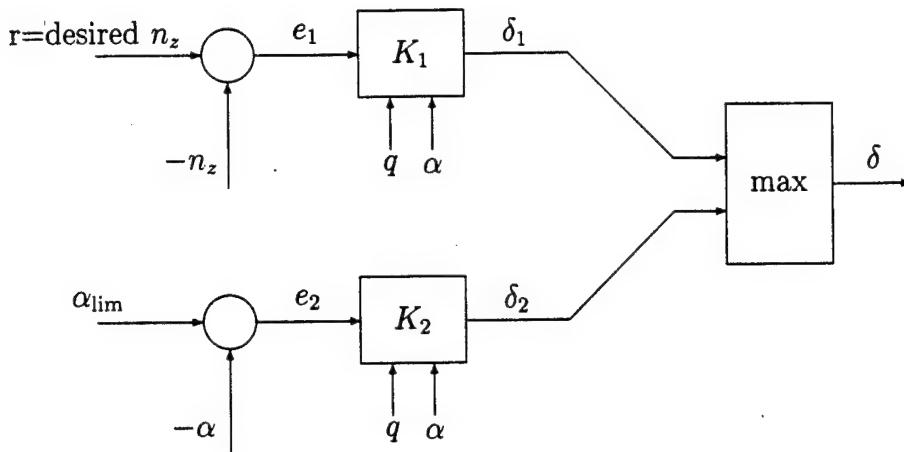
Figure 9-2: The Max Control System.

of this control system. First, we limit ourselves to the case where both controllers are implemented with full-state feedback. We discuss the well-posedness of the system and show an example output run. Next, we consider the equilibrium points of the system and their stability in the case where the pilot's reference input (desired normal acceleration) is clamped to zero. More practically, we answer the question, What is the behavior of this control system if the pilot lets go of the control stick?

§9.2.1 PRELIMINARY ANALYSIS OF THE EXAMPLE SYSTEM

First note that in our example system, the pair (A, B) is controllable. To make some headway, we restrict ourselves to the special case where the controllers K_1 and K_2 take the form of full-state feedback plus an offset term (for nonzero outputs):

$$\begin{aligned}
\delta_1 &= -Fx + [C_1(-A + BF)^{-1}B]^{-1}\alpha_{\text{lim}}, \\
\delta_2 &= -Gx + [(C_2 - D_2G)(-A + BG)^{-1}B + D_2]^{-1}r.
\end{aligned}$$

For convenience, we let

$$\begin{aligned}
k_1 &= [C_1(-A + BF)^{-1}B]^{-1}, \\
k_2 &= [(C_2 - D_2G)(-A + BG)^{-1}B + D_2]^{-1}.
\end{aligned}$$

Such constants generally need not exist. However, for our system we are keenly interested in the existence of k_1.[1] We have the following

Fact 9.1 *The constant k_1 is guaranteed to exist for our system whenever F is chosen such that $(A - BF)$ is stable.*[2]

Proof. (See Appendix 9.5.)   □

---

[1] We assume k_2 exists since it does not affect our later analysis, which is in the case $r = 0$.

[2] We say a matrix is **stable** when all its eigenvalues are strictly in the left-half plane.

Thus, the resulting max control law exists. It is simply

$$\delta = \max(-Fx + k_1\alpha_{\text{lim}}, -Gx + k_2r). \tag{9.1}$$

To get a feel for how the example system behaves, we have included some simulations. Figure 9-3 shows an example run of the just the tracking portion of the control system ($\alpha_{\text{lim}} = 0.6$, F chosen to place the closed-loop poles at -6 and -7). Part (a) shows normal acceleration tracks the desired trajectory well; (b) shows the α_{lim} constraint is violated to achieve this tracking.

Figure 9-4 shows the outputs when the full max control system is activated (with both $F = G$ chosen as F above). One easily sees that the controller acts as expected: it tracks the desired command well, except in that portion where tracking the command requires that the α_{lim} constraint be violated. In this portion, α is kept close to its constraint value (the maximum value of α in this simulation run was 0.6092).

§9.2.2 ANALYSIS FOR THE CASE $R \equiv 0$

The first thing we do is examine stability of Σ using the max control law in the case where $r \equiv 0$.

> NOTE. Similar analysis holds for r any constant after change of variables. If r is given by an asymptotically stable differential equation, then a theorem of [142] for triangular systems may be used with our result to conclude global asymptotic stability.

In the case $r \equiv 0$, the closed-loop system equations are then

$$\begin{aligned} \dot{x} &= Ax + B\max(-Fx + k_1\alpha_{\text{lim}}, -Gx) \\ &= (A - BG)x + B\max((G - F)x + k_1\alpha_{\text{lim}}, 0). \end{aligned}$$

In our analysis below, we suppose that we have done a reasonable job in designing the feedback controls F and G. That is, we assume $(A - BF)$ and $(A - BG)$ are stable. This is possible because (A, B) controllable.

Now, recall that (A, B) controllable implies that $(A - BG, B)$ is controllable. Thus, it suffices to analyze the following.

Definition 9.2 *The* **max system** *is defined by*

$$\Sigma_{\text{max}}: \qquad \dot{z} = Az + B\max(Fz + \gamma, 0),$$

where A and $A + BF$ are stable and (A, B) is controllable and $\gamma = k_1\alpha_{\text{lim}}$.

To fix ideas, let's look at simulation results. Figure 9-5 shows a max system trajectory with

$$A = \begin{bmatrix} -0.1 & 1 \\ -1 & -0.1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad F = \begin{bmatrix} -9 & 0 \end{bmatrix}, \qquad \gamma = -1.$$

Figure 9-7[3] shows the trajectory resulting from the same initial condition for the system $\dot{x} = Ax$; Figure 9-8 for the system $\dot{x} = (A + BF)x$. Both component systems are stable.

---

[3]We intentionally skipped a figure to allow better figure placement for comparison.

To simplify analysis of the max system, we can make a change of basis $(x = Pz)$, yielding

$$
\begin{aligned}
\dot{x} &= P\dot{z} \\
&= PAz + PB \max(Fz + \gamma, 0) \\
&= PAP^{-1}x + PB \max(FP^{-1}x + \gamma, 0),
\end{aligned}
$$

where P is any nonsingular matrix. In particular, P can be chosen so that the matrices PAP^{-1} and PB are in the so-called controller canonical form:

$$
PAP^{-1} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}, \tag{9.2}
$$

$$
PB = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{9.3}
$$

Note that $a_i > 0$ since PAP^{-1} is a stable matrix. Renaming matrices again, we have

Fact 9.3 *The max system $\Sigma_{\max}$ can be reduced to the system:*

$$
\dot{x} = Ax + B \max(Fx + \gamma, 0),
$$

where A and $A + BF$ are stable, (A, B) is controllable, and the matrices A and B are in controller canonical form.

We can do one more thing to simplify the max system just derived: expand the equations using the fact that A and B have controller canonical form. Doing this—and some equilibrium point analysis—we obtain

Remark 9.4 *The max system can be reduced to the following canonical form, denoted* **canonical max system**:

1.

$$
\begin{aligned}
\dot{x} &= y, \\
\dot{y} &= -ax - by + \max(fx + gy + \gamma, 0),
\end{aligned}
$$

where a, b, $a - f$, and $b - g$ are greater than zero.

2. Further, without loss of generality, we may assume that $\gamma \leq 0$, in which case the only equilibrium point of this system is the origin.

Proof.   The first part is a straightforward calculation, with the inequalities on the constants arising from the assumed stability of A and $A + BF$.

Now, let's analyze the equilibrium points of the canonical max system. The relevant equations are $y = 0$ and $ax = \max(fx + \gamma, 0)$. This second one must be analyzed in two cases:

$$
\begin{aligned}
ax &= 0, & fx + \gamma &\leq 0, \\
ax &= fx + \gamma, & fx + \gamma &> 0.
\end{aligned}
$$

Thus, $(0, 0)$ is an equilibrium point if and only if $\gamma \leq 0$; $(\gamma/(a - f), 0)$ is an equilibrium

point if and only if

$$\frac{f}{a-f}\gamma + \gamma \;>\; 0,$$
$$\frac{a}{a-f}\gamma \;>\; 0,$$
$$\gamma \;>\; 0,$$

where the last line follows from a and $a - f$ greater than zero. Therefore, the canonical max system has exactly one equilibrium point.

Finally, if $\gamma > 0$, changing coordinates to $z = x - \gamma/(a - f)$ yields $\dot{z} = y$ and

$$
\begin{aligned}
\dot{y} &= -a\left(z + \frac{\gamma}{a-f}\right) - by + \max\left(f\left(z + \frac{\gamma}{a-f}\right) + gy + \gamma, 0\right) \\
&= -az - by - \frac{a\gamma}{a-f} + \max\left(fz + gy + \frac{a\gamma}{a-f}, 0\right) \\
&= -az - by + fz + gy + \max\left(0, -fz - gy - \frac{a\gamma}{a-f}\right) \\
&= -(a-f)z - (b-g)y + \max\left((-f)z + (-g)y + \left(-\frac{a\gamma}{a-f}\right), 0\right).
\end{aligned}
$$

Now introducing new variables for the constants in parentheses, we obtain

$$
\begin{aligned}
\dot{z} &= y, \\
\dot{y} &= -\tilde{a}z - \tilde{b}y + \max(\tilde{f}z + \tilde{g}y + \tilde{\gamma}, 0).
\end{aligned}
$$

It is easy to check that the new variables satisfy the inequalities of the canonical form. Further, we have $\tilde{\gamma} < 0$, and thus $(0,0)$ the only equilibrium. $\square$

Next, note that this is equivalent to the second-order system:

$$\ddot{x} = -ax - b\dot{x} + \max(fx + g\dot{x} + \gamma, 0),$$

which we use below.

We have the following global results for the max system in the case where the reference input, r, is zero:

1. Limit cycles don't exist. Our max system consists of a logical (though Lipschitz continuous) switching between two stable linear systems, both of which admit negative divergence in their respective regions. Therefore, by Theorem 8.1, no limit cycles can exist.

2. The system is globally asymptotically stable. The proof is detailed below.

To prove global asymptotic stability, we first show

Remark 9.5 *The following is a Lyapunov function for the canonical max system:*

$$
\begin{aligned}
V &= \frac{1}{2}\dot{x}^2 + \int_0^x [a\xi - \max(f\xi + \gamma, 0)]d\xi \\
&\equiv \frac{1}{2}\dot{x}^2 + \int_0^x c(\xi)d\xi.
\end{aligned}
$$

Proof.   The proof has two major parts: (i) V is a positive definite (p.d.) function, and (ii) $\dot{V} \leq 0$.

(i) To show that V is a p.d. function, it is enough to show that $xc(x) > 0$ when $x \neq 0$ and $c(0) = 0$. The second fact follows from $\gamma \leq 0$. Computing

$$
\begin{aligned}
xc(x) &= ax^2 - x\max(fx + \gamma, 0) \\
&= \begin{cases} ax^2, & fx + \gamma \leq 0, \\ ax^2 - fx^2 - \gamma x, & fx + \gamma > 0. \end{cases}
\end{aligned}
$$

That the desired condition holds in the first case follows immediately from $a > 0$. For the second case, we consider

1. $x > 0$:
$$ ax^2 - fx^2 - \gamma x = (a - f)x^2 + (-\gamma)x > 0 + 0 = 0, $$

2. $x < 0$:
$$ ax^2 - fx^2 - \gamma x = ax^2 + (-x)(fx + \gamma) > 0 + 0 = 0. $$

Thus V is a p.d. function.

(ii) Next, we wish to show that $\dot{V} \leq 0$. To that end, we compute

$$
\begin{aligned}
\dot{V} &= \dot{x}\ddot{x} + c(x)\dot{x} \\
&= \dot{x}[-ax - b\dot{x} + \max(fx + g\dot{x} + \gamma, 0)] + ax\dot{x} - \max(fx + \gamma, 0)\dot{x} \\
&= -b\dot{x}^2 + \dot{x}\max(fx + g\dot{x} + \gamma, 0) - \dot{x}\max(fx + \gamma, 0).
\end{aligned}
$$

Now, there are four cases to be dealt with:

1. If $fx + g\dot{x} + \gamma \leq 0$ and $fx + \gamma \leq 0$, then $\dot{V} = -b\dot{x}^2 \leq 0$.
2. If $fx + g\dot{x} + \gamma > 0$ and $fx + \gamma > 0$, then $\dot{V} = -(b - g)\dot{x}^2 \leq 0$.
3. If $fx + g\dot{x} + \gamma \leq 0$ and $fx + \gamma > 0$, then

$$ \dot{V} = -b\dot{x}^2 - \dot{x}(fx + \gamma). $$

If $\dot{x} \geq 0$, then $\dot{V} \leq 0$. If $\dot{x} < 0$, then, using $(b - g) > 0$, we obtain

$$
\begin{aligned}
b &> g \\
b\dot{x} &< g\dot{x}, \\
fx + \gamma + b\dot{x} &< fx + \gamma + g\dot{x}, \\
fx + \gamma + b\dot{x} &< 0, \\
-\dot{x}[fx + \gamma + b\dot{x}] &< 0, \\
\dot{V} &< 0.
\end{aligned}
$$

4. If $fx + g\dot{x} + \gamma > 0$ and $fx + \gamma \leq 0$, then

$$ \dot{V} = -b\dot{x}^2 + \dot{x}(fx + \dot{x}g + \gamma). $$

If $\dot{x} \leq 0$, then $\dot{V} \leq 0$. If $\dot{x} > 0$,

$$\dot{V} = -(b-g)\dot{x}^2 + \dot{x}(fx + \gamma) \leq 0.$$

$\square$

Global asymptotic stability results from the facts that (1) the origin is the only invariant set for which $\dot{V} = 0$ and (2) $V(x) \to \infty$ as $\|x\| \to \infty$ [132].

§9.3 EXAMPLE 2: THE MAX SYSTEM CONTINUATION

In this section we analyze a variant of the max system introduced in §9.2. Specifically, recall that the max system can be reduced to the canonical form of Remark 9.4:

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -ax - by + \max(fx + gy + \gamma, 0), \end{aligned}$$

where a, b, $a - f$, and $b - g$ are greater than zero and $\gamma \leq 0$. It was shown in §9.2 that the only equilibrium point of this system is the origin, which is globally asymptotically stable.

We now examine a *continuation* of the max system that uses a differential equation to "dynamically smooth" the max function.

Definition 9.6 *The* **max system continuation** *is defined by*

$$\begin{aligned} \dot{x} &= Ax + B\delta, \\ \dot{\delta} &= \alpha^2[\max(Fx + \gamma, 0) - \delta], \end{aligned}$$

where A and $A + BF$ are stable and (A, B) is controllable. Also, $\gamma = k_1\alpha_{\lim}$ and $\alpha \neq 0$.

This equation represents a smoothing of the max function's output; it provides a type of dynamic hysteresis that smooths transitions. Note also that this equation represents a singular perturbation of the original max system. It can be used to model the fact that the elevator angle does not exactly track the desired control trajectory specified by the max function. To compare the max and max continuation systems, consider Figure 9-6. This figure shows the continuation of the max system trajectory of Figure 9-5 with $\alpha^2 = 16$ and $\delta(0) = \max(Fx(0) + \gamma, 0)$. Note that, compared with the original max system, the switching is "delayed" and the trajectories are smoother, as expected.

By changing basis with the matrix $T = \text{blockdiag}\{P, 1\}$, where P is chosen so that the matrices PAP^{-1} and PB are in the so-called controller canonical form (see Equations (9.2) and (9.3)), we obtain

Definition 9.7 *The max system continuation can be reduced to the following canonical form, denoted* **canonical max system continuation***:*

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -ax - by + \delta, \\ \dot{\delta} &= \alpha^2[\max(fx + gy + \gamma, 0) - \delta], \end{aligned}$$

subject to initial conditions

$$x(0) = x_0, \qquad y(0) = y_0, \qquad \delta(0) = \max(fx_0 + gy_0 + \gamma, 0),$$

where a, b, $a - f$, and $b - g$, are greater than zero; $\gamma \leq 0$; and $\alpha \neq 0$.

This is the system we study in the remainder of this section. Note the added constraint on the initial condition of δ.

> NOTE. The constraint on $\delta(0)$ is for convenience. It can be relaxed to be within ϵ of this value (where ϵ arises in our proofs below), with the same analytical results holding true. Specifically, Fact 9.9 still holds when Equation (9.4) is replaced by $|\delta(0) - \max(fx(0) + gy(0) + \gamma, 0)| < \epsilon$.

Remark 9.8 *The only equilibrium point of this system is the origin, which is locally asymptotically stable (when $\gamma < 0$).*

Proof. From the first two equations, we have the constraints $y = 0$ and $\delta = ax$. From the last one we obtain the following two cases:

1. $fx + \gamma \leq 0$: $-ax = 0$, which implies $x = \delta = 0$.

2. $fx + \gamma > 0$: $(-a + f)x + \gamma = 0$, which implies $x = \gamma/(a - f)$. However, this can't occur since
$$fx + \gamma = \frac{f\gamma}{(a - f)} + \gamma = \frac{a\gamma}{(a - f)} \leq 0.$$

The origin is locally asymptotically stable because it is a linear system in some neighborhood of the origin (since $\gamma < 0$). □

This system is globally asymptotically stable when $f = g = 0$, because it reduces to a stable linear system in this case. For the special case where $\gamma = 0$, both component linear systems can be made stable by choosing α large enough. However, this in itself does not imply that the whole system is stable. We say more about this case at the end of the section.

The rest of this section explores the stability of the max system continuation by using Lemma 8.4.

§9.3.1 ASYMPTOTIC STABILITY WITHIN ARBITRARY COMPACT SETS

In this subsection we show that the max system continuation can be made asymptotically stable to the origin—within an arbitrary compact set containing the origin—by choosing the parameter α large enough.

> **Important Note.** Since δ is subject to initial conditions depending on x and y (see Definition 9.6), this stability is with respect to arbitrary sets of initial conditions for *x and y only.*

This subsection only considers the case $\gamma < 0$. In this case, the plane $fx + gy + \gamma = 0$ is a positive distance, call it d, away from the origin. Further, the three-dimensional linear system associated with the max system continuation about the origin with matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ -a & -b & 1 \\ 0 & 0 & -\alpha^2 \end{bmatrix}$$

is asymptotically stable, so there is some (open) ball around the origin (in $\mathbb{R}^3$) of radius $\Delta_s \leq d$ such that once a trajectory of the max system continuation enters the ball of radius Δ_s, it tends toward the origin. Similarly, the max system is an asymptotically stable linear system near the origin, so there is some ball around the origin (in $\mathbb{R}^2$) of radius $\Delta_m \leq d$ such that once a solution to the max system enters the ball of radius Δ_m, it tends towards the origin. For convenience, define $\Delta = \min(\Delta_m, \Delta_s)$.

Now, note that the max system and max system continuation can be written in the form required by Lemma 8.4 by choosing

$$A = \begin{bmatrix} 0 & 1 \\ -a & -b \end{bmatrix},$$

i.e.,

$$\begin{aligned} F(x, y, t) &= \max(fx + gy + \gamma, 0), \\ G(x, y, t) &= \delta, \end{aligned}$$

where $\dot{\delta} = \alpha^2 [\max(fx + gy + \gamma, 0) - \delta]$. An important fact is the following:

Fact 9.9 *Given*

$$\delta(0) = \max(fx(0) + gy(0) + \gamma, 0) \tag{9.4}$$

and $\epsilon > 0$, we can choose α large enough so that

$$|\delta(t) - \max(fx(t) + gy(t) + \gamma, 0)| < \epsilon, \qquad t \geq 0.$$

Proof.   Since $\max(fx + gy + \gamma, 0)$ is Lipschitz continuous, we can apply Lemma 8.7 with $u(\cdot) \equiv \max(fx(\cdot) + gy(\cdot) + \gamma, 0)$, $x(\cdot) \equiv \delta(\cdot)$, and $\epsilon_0 = 0$.   □

Below, let $\mu(t)$ and $\sigma(t)$ represent solutions to the max and max continuation systems, respectively. Next, consider the projection operator

$$\begin{aligned} \pi : \mathbb{R}^3 &\rightarrow \mathbb{R}^2, \\ \pi\left([x, y, \delta]^T\right) &= [x, y]^T. \end{aligned}$$

Remark 9.10 *If $\gamma < 0$, the max system continuation can be made asymptotically stable to the origin within an arbitrary compact set containing it by choosing the parameter α large enough.*

Proof.   First, pick any compact set, Ω, containing the origin (of the max system). Next, we examine the trajectories of the max and max continuation systems from an arbitrary initial condition, $\mu_0 \in \Omega$. Recall that δ_0, and hence σ_0, is completely determined by μ_0. In particular, $\delta_0 = \max(fx_0 + gy_0 + \gamma, 0)$ and $\pi(\sigma_0) = \mu_0$.

Since the max system is globally asymptotically stable, there is a time, $T(\mu_0, \Delta)$, such that for $t > T$, we have $\|\mu(t)\| < \Delta/3$. Thus, we have $\max(fx(t) + gy(t) + \gamma, 0) = 0$ for all $t > T$. Now, according to Fact 9.9 we can pick α large enough so that

$$\epsilon < \min\left(\frac{\Delta}{3}, \frac{\Delta(\eta + cL)}{3c\left(e^{(\eta + cL)T} - 1\right)}\right). \tag{9.5}$$

At this point, we have, from Lemma 8.4,

$$\|\mu(T) - \pi(\sigma(T))\| \leq \frac{\epsilon c}{\eta + cL} \left(e^{(\eta + cL)T} - 1\right) \leq \frac{\Delta}{3}.$$

Now, by construction we have $|\delta| < \Delta/3$. Thus, we have $\|\sigma(T)\| < \Delta$. From this point on, $\sigma(t)$ tends asymptotically toward the origin.

Finally, since Ω is compact, there is a finite time $\tau \geq T(\mu_0, \Delta)$ for all $\mu_0 \in \Omega$. Thus, we can pick ϵ (and then α) to achieve the desired inequality for all initial conditions. $\quad\square$

Note that if $\eta + cL < 0$, then ϵ—and hence α—can be chosen constant for all T. On the other hand, if $\eta + cL > 0$, restrictions on the magnitude of α may only guarantee asymptotic stability within some finite distance from the origin.

It is also important to realize that the same analysis holds for any other dynamic or nondynamic continuous variable used to approximate the max function, if it is such that it can be kept within ϵ of the max function for arbitrary ϵ. (Also recall the note on p. 145.)

§9.3.2 THE CASE $\gamma = 0$

For the special case $\gamma = 0$, the max system continuation represents a switching between the following component linear systems:

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ -a & -b & 1 \\ 0 & 0 & -\alpha^2 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 1 & 0 \\ -a & -b & 1 \\ f\alpha^2 & g\alpha^2 & -\alpha^2 \end{bmatrix}.$$

Remark 9.11 *Both component linear systems can be made stable by choosing α large enough.*

Proof. (See Appendix 9.5.) $\quad\square$

Thus the component linear systems of the max system continuation with $\gamma = 0$ can be chosen so both are stable. However, this need not imply that the whole system is stable.

> NOTE. A counterexample can be constructed after one that appears in [139]: Use the asymptotically stable systems of Figures 9-7 and 9-8 (Systems I and II, respectively), activating System II in quadrants 4 and 2, System I in quadrants 3 and 1. Also, see Example 8.9.

The comparison arguments of the previous subsection do not apply now since we cannot find a Δ like we did there. Thus, we can only use Lemma 8.4 to get bounds on the trajectories of the max system continuation. Note, however, that if $\eta + cL < 0$ then global asymptotic stability of the max system implies ultimate boundedness of the max system continuation.

One may be able to say more about specific instances of the max system continuation (i.e., knowledge of the constants). For example, some cases may yield to our robustness of linear ODEs lemma by comparing the case $\gamma = 0$ with $\gamma = -\epsilon$. Alternatively, one could invoke robustness results in the literature, e.g., [11, Theorem 6.1]. These tools can't be invoked in the general case because, roughly, the parameter α affects both η and L in conflicting fashion.
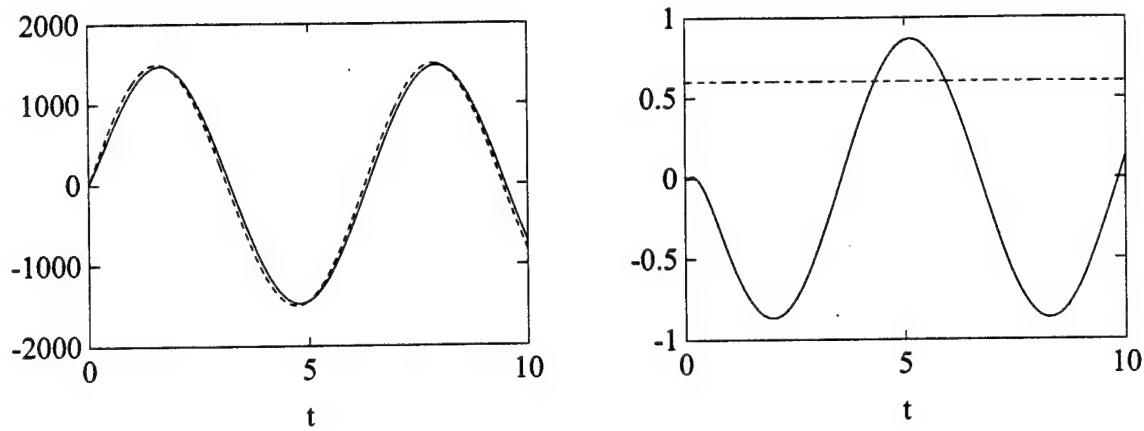
Figure 9-3: Outputs of the tracking controller: (a) normal acceleration, n_z (solid), and desired normal acceleration, r (dashed); (b) angle of attack, α (solid), and α's limit (dashed).
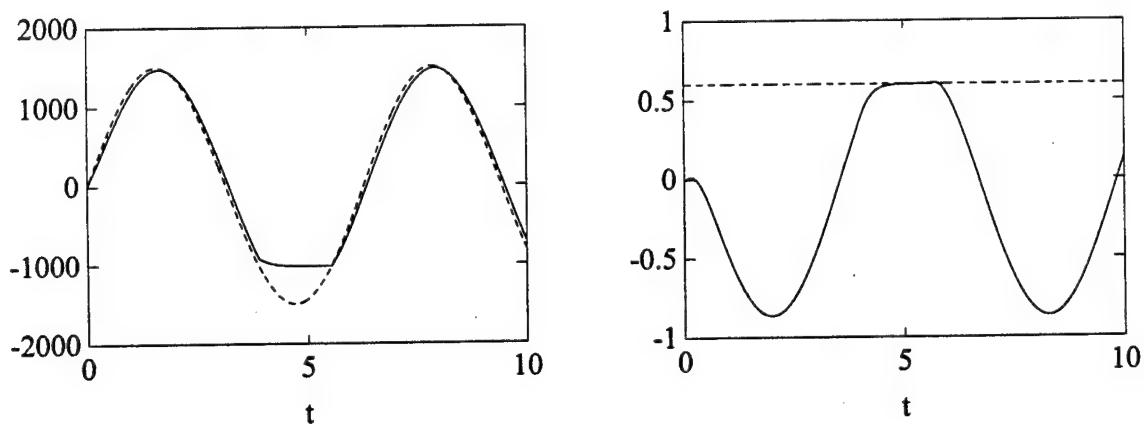
Figure 9-4: Outputs of the max controller: (a) normal acceleration, n_z (solid), and desired normal acceleration, r (dashed); (b) angle of attack, α (solid), and α's limit (dashed).
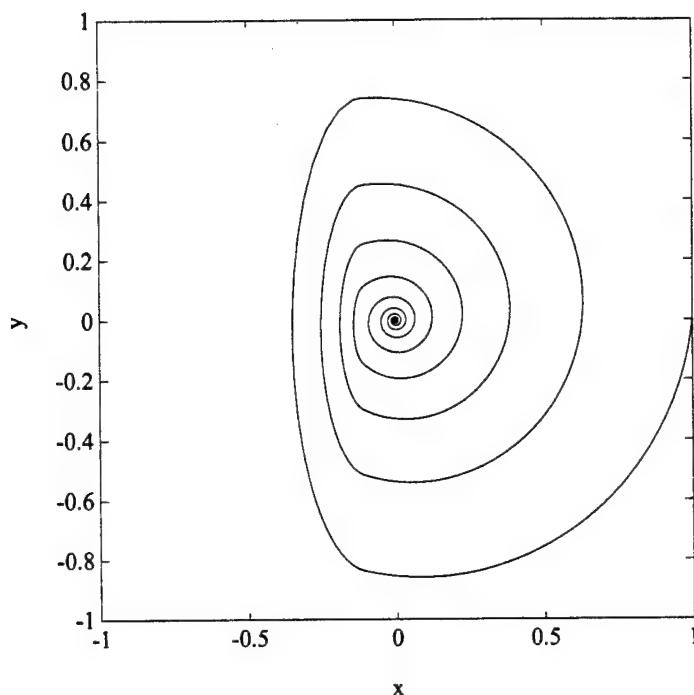
Figure 9-5: Max System Trajectory.
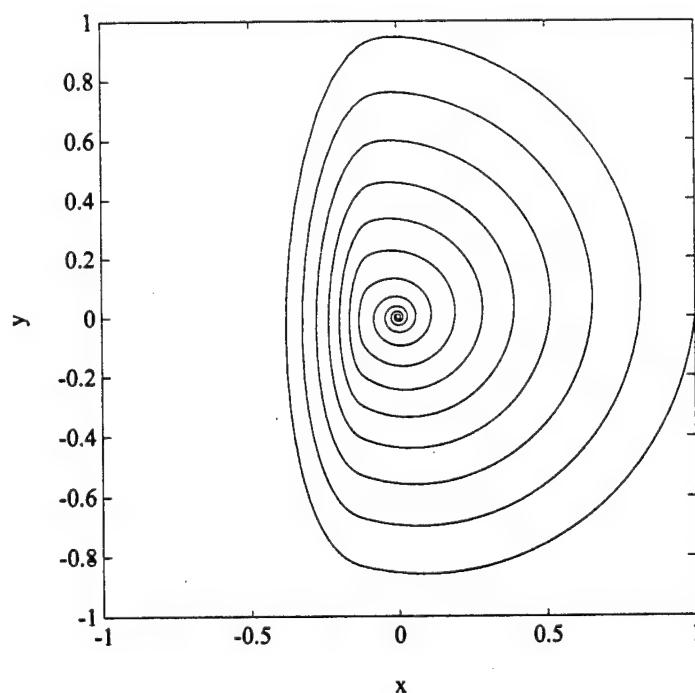
Figure 9-6: Max System Continuation Trajectory.
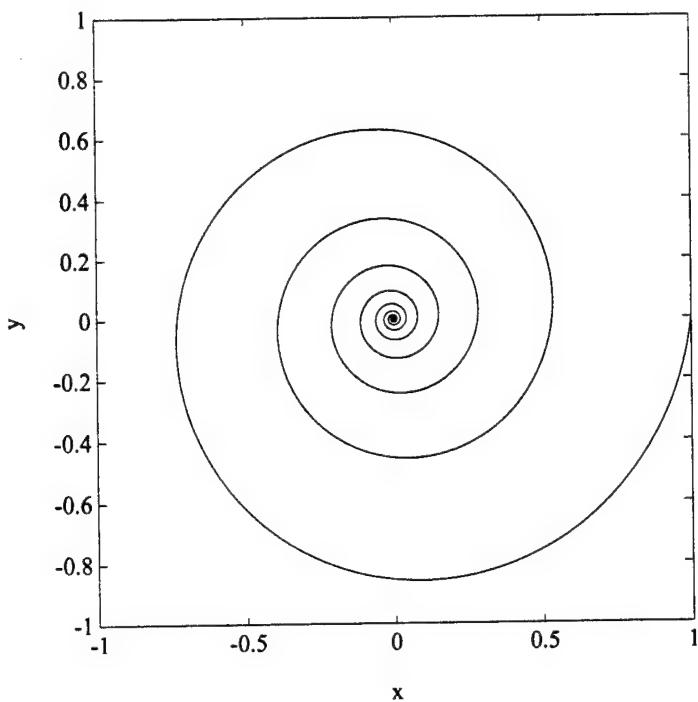
Figure 9-7: A System Trajectory.
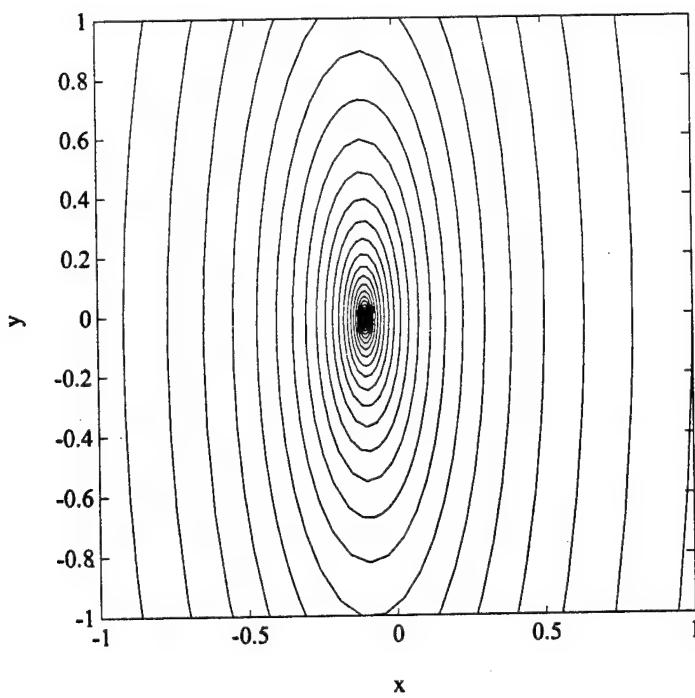
Figure 9-8: $A + BF$ System Trajectory.

§**9.4 NOTES**

The work in this chapter first appeared in [21], later summarized in [28]. Prof. Gunter Stein supplied the max system problem.

§**9.5 APPENDIX**

§**9.5.1 MAX SYSTEM**

Proof. [of Fact 9.1] We first need the following theorem [92, Theorem 3.10]:

Theorem 9.12 (Kwakernaak and Sivan) *Consider the time-invariant system*

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$z(t) = Dx(t),$$

where z and u have the same dimensions. Consider any asymptotically stable time-invariant control law

$$u(t) = -Fx(t) + u'(t).$$

Let $H(s)$ be the open-loop transfer matrix

$$H(s) = D(sI - A)^{-1}B,$$

and $H_c(s)$ the closed-loop transfer matrix

$$H_c(s) = D(sI - A + BF)^{-1}B.$$

Then $H_c(0)$ is nonsingular and the controlled variable $z(t)$ can under steady-state conditions be maintained at any constant value z_0 by choosing

$$u'(t) = H_c^{-1}(0)z_0$$

if and only if $H(s)$ has a nonzero numerator polynomial that has no zeroes at the origin.

For our system, we have

$$C_1(sI - A)^{-1}B = \frac{0.1(s - 9)}{s^2 + 2s + 11},$$

which has a nonzero numerator with no zeroes at the origin. □

§**9.5.2 MAX SYSTEM CONTINUATION**

Proof. [of Remark 9.11] A_1 is stable because it is upper block triangular with stable blocks. One can check that the characteristic polynomial of A_2 is

$$\lambda^3 + (b + \alpha^2)\lambda^2 + [a + \alpha^2(b - g)]\lambda + [\alpha^2(a - f)] \equiv \lambda^3 + \alpha'\lambda^2 + \beta'\lambda + \gamma',$$

The Routh test (to verify stable roots) reduces here to [130, p. 175]:

1. $\alpha', \beta', \gamma' > 0$,

2. $\beta' > \gamma'/\alpha'$.

The first of these is verified by our conditions on a, b, f, g, and α. The second says we need

$$a + \alpha^2(b - g) > \frac{\alpha^2(a - f)}{b + \alpha^2},$$

which reduces to

$$\alpha^4(b - g) + \alpha^2[b(b - g) + f] + ab > 0,$$
$$\alpha^4 + \alpha^2\left[b + \frac{f}{(b - g)}\right] + \frac{ab}{(b - g)} > 0.$$

Since the last term on the left-hand is positive by our previous conditions (a, b, and $b - g$ greater than zero) it is sufficient for

$$\alpha^4 + \alpha^2\left[b + \frac{f}{(b - g)}\right] > 0,$$
$$\alpha^2 + b + \frac{f}{(b - g)} > 0.$$

Again, since $b > 0$, it is sufficient for

$$\alpha^2 + \frac{f}{(b - g)} > 0,$$
$$\alpha^2 > \frac{-f}{(b - g)}.$$

$\square$

Part III

Control of Hybrid Systems

Chapter 10

Control of Hybrid Systems: Theoretical Results

In this chapter we define an optimal control problem in our unified hybrid control framework and derive some theoretical results. The necessity of our assumptions—or ones like them—are demonstrated using examples throughout the sequel. The main results are as follows: The existence of optimal and near-optimal controls for the problem are established. The "value function" associated with this problem is expected to satisfy a set of *generalized quasi-variational inequalities* (*GQVIs*), which are formally derived. We conclude with a brief list of some of the more striking open issues.

§10.1 THE CONTROL PROBLEM

In this section, we define a control problem and elucidate all assumptions used in deriving the results of the sequel.

§10.1.1 PROBLEM

Let $a > 0$ be a **discount factor**. We add to our previous model the following known maps:

1. **Running cost** $k_i : X_i \times X_i \times U \to \mathbb{R}_+$.

2. **Autonomous jump cost** or **transition cost** $c_{a,i} : A_i \times V \to \mathbb{R}_+$. We may define

3. **Controlled jump cost** or **impulse cost** $c_{c,i} : C_i \times D \to \mathbb{R}_+$, satisfying for all $i, j \in \mathbb{Z}_+$ the conditions

$$c_c(x,y) \geq c_0 > 0, \qquad \forall x \in C_i, \, y \in D, \qquad (10.1)$$

$$c_c(x,y) < c_c(x,z) + e^{-a\Delta_c(x,z)}c_c(z,y), \quad \forall x \in C_i, \, z \in D \cap C_j, \, y \in D. \quad (10.2)$$

Important Note. As before, we have used the shorthand $c_c : C \times D \to \mathbb{R}_+$, defined in the obvious way. However, we have suppressed reference to the index state, since it is obvious. We do the same below, with $c_a(x(t), y) \equiv c_{a,i}(x(t), y)$ if $x(t) \in X_i$. In such a case, it is equivalent to think of $x(t)$ as a member of the **formal union** $\{X_i\}_{i=0}^{\infty}$. Thus, to ease notation we use this shorthand throughout for the maps G, Δ_a, Δ_c, c_a, c_c, etc. To alert the reader, such formal unions are denoted using the symbol $\bigsqcup$.

Thus, autonomous jumps are done at a cost of $c_a(x(\tau_j), v_j)$ paid at time τ_j; controlled jumps at a cost of $c_c(x(\tau_j), x(\Gamma_j))$ paid at time τ_j.

In addition to the costs associated with the jumps as above, the controller also incurs a running cost of $k_i(x(t), x[t], u(t))$ per unit time during the intervals $[\Gamma_{j-1}, \tau_j)$, $j \in \mathbb{Z}_+$. The **total discounted cost** is defined as

$$\int_{\mathbb{T}} e^{-at} k_i(x(t), x[t], u(t)) \, dt + \sum_i e^{-a\sigma_i} c_a(x(\sigma_i), v_i) + \sum_i e^{-a\zeta_i} c_c(x(\zeta_i), x(\zeta_i')), \qquad (10.3)$$

where $\mathbb{T} = \mathbb{R}_+ \setminus (\bigcup_i [\tau_i, \Gamma_i])$, $\{\sigma_i\}$ (respectively $\{\zeta_i\}$) are the successive pre-jump times for autonomous (respectively controlled) jumps and ζ_j' is the post-jump time for the jth controlled jump. The **decision** or **control** variables over which Equation (10.3) is to be minimized are

- the **continuous control** $u(\cdot)$,

- the **discrete controls** $\{v_i\}$, exercised at the pre-jump times of autonomous jumps,

- the pre-jump or **intervention times** $\{\zeta_i\}$ of controlled jumps, and the associated **destinations** $\{x(\zeta_i')\}$.

As for the periods $[\tau_j, \Gamma_j)$, we shall follow the convention that the system remains frozen during these intervals. Note that Equation (10.1) rules out from consideration infinitely many controlled jumps in a finite interval and Equation (10.2) rules out the merging of post-jump time of a controlled jump with the pre-jump time of the next controlled jump.

Our framework clearly includes conventional impulse control [14].

§10.1.2 ASSUMPTIONS

Throughout the sequel, we make use of the following further assumptions on our abstract model, which are collected here for clarity and convenience.

For each $i \in \mathbb{Z}_+$, the following hold: X_i is the closure of a connected open subset of Euclidean space $\mathbb{R}^{d_i}$, $d_i \in \mathbb{Z}_+$, with Lipschitz boundary ∂X_i. $A_i, C_i, D_i \subset X_i$ are closed. In addition, ∂A_i is Lipschitz and contains ∂X_i.

The maps G, Δ_a, Δ_c, c_a, and c_c are bounded uniformly continuous; the k_i are uniformly bounded and uniformly equicontinuous. The vector fields f_i, $i \in \mathbb{Z}_+$, are bounded (uniformly in i), uniformly Lipschitz continuous in the first argument, uniformly equicontinuous with respect to the rest. U, V are compact metric spaces. Below, $u(\cdot)$ is a U-valued control process, assumed to be measurable.

All the above are fairly mild assumptions. The following are more technical assumptions. They may be traded for others as discussed in §10.4. However, in the sequel we construct examples pointing out the necessity of such assumptions or ones like them.

Assumption 10.1 $d(A_i, C_i) > 0$ and $\inf_{i \in \mathbb{Z}_+} d(A_i, D_i) > 0$, d being the appropriate Euclidean distance.

Assumption 10.2 Each D_i is bounded and for each i, there exists an integer $N(i) < \infty$ such that for $x \in C_i$, $y \in D_j$, $j > N(i)$, $c_c(x, y) > \sup_z J(z)$.

Assumption 10.3 *For each i, ∂A_i is an oriented C^1-manifold without boundary and at each point x on ∂A_i, $f_i(x,z,u)$ is "transversal" to ∂A_i for all choices of z, u. By this we require that (1) the flow lines be transversal in the usual sense[1] and (2) the vector field does not vanish on ∂A_i.*

Assumption 10.4 *Same as Assumption 10.3 but with C_i replacing A_i.*

§10.2 EXISTENCE OF OPTIMAL CONTROLS

Let $J(x)$ denote the infimum of Equation (10.3) over all choices of $u(\cdot), \{v_i\}, \{\zeta_i\}, \{x(\zeta_i')\}$ when $x(0) = x$. We have

Theorem 10.5 *A finite optimal cost exists for any initial condition.*

Proof.   Let F, K, Q be bounds of the f_i, k_i, and c_a, respectively. Then, choosing to make no controlled jumps and using arbitrary u, v we have that

$$J(x) \leq K \int_0^\infty e^{-at}\, dt + \sum_i e^{-a\sigma_i} Q \leq K/a + Q \sum_i e^{-a\sigma_i}.$$

Let $\beta = \inf_{i \in \mathbb{Z}_+} d(A_i, D_i)$. Then $\sigma_{i+1} - \sigma_i \geq \beta/F$, so the second term is bounded by $Q \sum_{i=1}^\infty (e^{-a\beta/F})^i$, which converges. $\square$

The following corollary is immediate from the argument above:

Corollary 10.6 *There are only finitely many autonomous jumps in finite time.*

To see why an assumption like Assumption 10.1 is necessary for the above results, one need only consider the following one-dimensional example:

Example 10.7 *Let $X_i = [0,2]$, $A_i = \{0,2\}$, and $f_i(\cdot,\cdot,\cdot) \equiv -1$ for each $i \in \mathbb{Z}_+$. Also for each i, define $C_i = \emptyset$, $D_i = 1/i^2$ and $G(A_i, \cdot) \equiv 1/(i+1)^2$. Finally, let $\Delta_a(\cdot,\cdot) \equiv 0$ and $c_a(\cdot,\cdot) \equiv 1$. Starting in X_1 at $x(0) = 1$, we see that*

$$x\left(\sum_{i=1}^N \frac{1}{i^2}\right) = \frac{1}{(N+1)^2}.$$

Since the sum of inverse squares converges, we will accumulate an infinite number of jumps and infinite cost by time $t = \pi^2/6$.

Next, we show that $J(x)$ is attained for all x if we extend the class of admissible $u(\cdot)$ to "relaxed" controls. The **relaxed control** framework [155] is as follows: We suppose that $U = \mathcal{P}(U')$, defined as the space of probability measures on a compact space U' with the topology of weak convergence [18]. Also

$$f_i(x,z,u) = \int f_i'(x,z,u)u(dy), \qquad i \in \mathbb{Z}_+,$$

$$k_i(x,z,u) = \int k_i'(x,z,u)u(dy), \qquad i \in \mathbb{Z}_+,$$

---

[1]Transversality implies that ∂A_i is $(d_i - 1)$-dimensional.

for suitable $\{f_i'\}$, $\{k_i'\}$ satisfying the appropriate continuity/Lipschitz continuity requirements. The relaxed control framework and its implications in control theory are well known and the reader is referred to [155] for details.

Theorem 10.8 *An optimal trajectory exists for any initial condition.*

Proof. (See Appendix 10.6.) □

It is easy to see why Theorem 10.8 may fail in absence of Assumption 10.2:

Example 10.9 *Suppose, for example, $k_i(x, z, u) \equiv \alpha_i$ and $c_a(x, v) \equiv \beta_i$ when $x \in X_i$, $c_c(x, y) \equiv \gamma_{i,j}$ when $x \in X_i$, $y \in X_j$, with α_i, β_i, $\gamma_{i,j}$ strictly decreasing with i, j. It is easy to conceive of a situation where the optimal choice would be to "jump to infinity" as fast as you can.*

The theorem may also fail in the absence of Assumption 10.3 as the following two-dimensional system shows:

Example 10.10

$$\begin{aligned}
\dot{x}_1(t) &= 1, & x_1(0) &= 0, \\
\dot{x}_2(t) &= u, & x_2(0) &= 0,
\end{aligned}$$

with $u \in [0, 1]$ and cost

$$\int_0^\infty e^{-t} \min\{|x_1(t) + x_2(t)|, 10^{20}\} \, dt,$$

with the provision that the trajectory jumps to $[10^{10}, 10^{10}]$ on hitting a certain curve A. For A, consider two possibilities:

1. *the line segment $\{x_1 = 1, -1 \leq x_2 \leq 0\}$, a C^1-manifold with boundary;*

2. *the circle $\{(x_1, x_2) \mid (x_1 - 1)^2 + (x_2 + 1)^2 = 1\}$, a C^1-manifold without boundary, but the vector field $(1, u)$ with $u = 0$ is not transversal to it at $(1, 0)$..*

It is easy to see that the optimal cost is not attained in either case.

Also, it is not enough that the flow lines for each control be transversal in the usual sense as the following one-dimensional example shows:

Example 10.11 *Let $X_1 = X_2 = \mathbb{R}_+$.*

$$f_1(x, y, u) = -x + u, \qquad f_2(x, y, u) = 0, \qquad u \in [-1, 0],$$

with running cost $\min\{K, |x|\}$ and $G_1(0, \cdot) \equiv (K, 2)$. Choosing, for example, $K > 1$ one sees that the optimal cost cannot be attained for any $1 \geq x(0) > 0$.

Coming back to the relaxed control framework, say that $u(\cdot)$ is a **precise** control if $u(\cdot) = \delta_{q(\cdot)}(dy)$ for a measurable $q : [0, \infty) \rightarrow U'$ where δ_z denotes the Dirac measure at $z \in U'$. Let M denote the set of measures on $[0, T] \times U'$ of the form $dt\, u(t, dy)$ where $u(\cdot)$ is a relaxed control, and M_0 its subset corresponding to precise controls. It is known that M_0 is dense in M with respect to the topology of weak convergence [155]. In conjunction with the additional assumption Assumption 10.4 below, this allows us to deduce the existence of ϵ-optimal control policies using precise $u(\cdot)$, for every $\epsilon > 0$.

Theorem 10.12 *Under Assumptions 10.2–10.4, for every $\epsilon > 0$ an ϵ-optimal control policy exists wherein $u(\cdot)$ is precise.*

Proof. (See Appendix 10.6.) □

Remarks. If $\{\overline{f_i}(x, z, y) \mid y \in U'\}$ are convex for each x, z, a standard selection theorem [155] allows us to replace $u^\infty(\cdot)$ by a precise control which is optimal. Even otherwise, using Caratheodory's theorem (which states that each point in a compact subset of $\mathbb{R}^n$ is expressible as a convex combination of at most $n + 1$ of its extreme points) and the aforementioned selection theorem, one may suppose that for $t \geq 0$, the support of $u^\infty(t)$ consists of at most $d_i + 1$ points when $x(t) \in X_i$.

§10.3 THE VALUE FUNCTION

In the foregoing, we had set $[0] = 0$ and thus $x[0] = x(0) = x_0$. More generally, for $x(0) = x_0 \in X_{i_0}$, we may consider $x[0] = y$ for some $y \in X_{i_0}$, making negligible difference in the foregoing analysis. Let $V(x, y)$ denote the optimal cost corresponding to this initial data. Then in dynamic programming parlance, $(x, y) \mapsto V(x, y)$ defines the "value function" for our control problem.

In view of Assumption 10.3, we can speak of the **right side** of ∂A_i as the side on which $f_i(\cdot, \cdot, \cdot)$ is directed towards ∂A_i, $i \in \mathbb{Z}_+$. A similar definition is possible for the right side of ∂C_i (in light of Assumption 10.4).

Definition 10.13 *Say that $(x_n, y_n) \to (x_\infty, y_\infty)$ **from the right** in $\bigsqcup_i (X_i \times X_i)$ if $y_n \to y_\infty$ and either $x_n \to x_\infty \notin \bigsqcup_i (\partial A_i \cup \partial C_i)$ or $x_n \to x_\infty \in \bigsqcup_i (\partial A_i \cup \partial C_i)$ from the right side.*

V is said to be **continuous from the right** if $(x_n, y_n) \to (x_\infty, y_\infty)$ from the right implies $V(x_n, y_n) \to V(x_\infty, y_\infty)$.

Theorem 10.14 *V is continuous from the right.*

Proof. (See Appendix 10.6.) □

Corollary 10.15 *V is continuous on $\bigsqcup_i (X_i \times X_i) \backslash [(\partial A_i \cup \partial C_i) \times X_i]$.*

Again, Example 10.11 shows the necessity of the vector field's not vanishing on ∂A_i. Unfortunately, V need not be continuous in the data of the hybrid system:

Example 10.16 *Let $X_1 = X_2 = \mathbb{R}$, $f_1 = f_2 \equiv 1$, $k_1 = 1$, and $k_2 = 0$. Further, let $A_1 = \{p\}$, $G_1(p) = (p, 2)$, and $c(p, p) = 0$. Then we have*

$$V(x) = \begin{cases} 0, & x \in X_2, \\ \left(1 - e^{-\alpha(p-x)}\right)/\alpha, & x \in X_1, \ x \leq p, \\ 1/\alpha, & x \in X_1, \ x > p. \end{cases}$$

It is clear that the optimal cost-to-go is not continuous at p in the autonomous jump set data A_1.

We shall now formally derive the *generalized quasi-variational inequalities* $V(\cdot, \cdot)$ is expected to satisfy. Let $C = \bigsqcup_i (C_i \times X_i)$ and $E \subset C$ the set on which

$$V(x, y) = \min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x, z)} V(z, z) \right\}, \tag{10.4}$$

where $i \in \mathbb{Z}_+$ is such that $x, y \in X_i$. For $(x, y) \in E$, if $x(t) = x$ and $x[t] = y$, an optimal decision (not necessarily the only one) would be to jump to a z where the minimum on the right hand side of Equation (10.4) is obtained. On the other hand, for $(x, y) \in C \backslash E$,

$$V(x, y) < \min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x,z)} V(z, z) \right\},$$

with i as above and it is not optimal to execute a controlled jump. For $x \in A_i$, however, an autonomous jump is mandatory and thus

$$V(x, y) = \min_v \left\{ c_a(x, v) + e^{-a\Delta_a(x,v)} V(G(x, v), G(x, v)) \right\}.$$

Suppose E is a closed subset of $\bigsqcup_i (X_i \times X_i)$. Let $H = E \sqcup (\bigsqcup_i (A_i \times X_i))$, with $M = (\bigsqcup_i (X_i \times X_i)) \backslash H$. Let $(x, y) \in M^o$, with $x, y \in X_{i_0}$ (say). Let O be a bounded open neighborhood of (x, y) in M^o with a smooth boundary ∂O and $\nu = \inf\{t \geq 0 \,|\, (x(t), y) \notin O\}$, where $x(\cdot)$ satisfies

$$\dot{x}(t) = f_{i_0}(x(t), y, u(t)), \qquad x(0) = x, t \in [0, \nu]. \tag{10.5}$$

Note that y is a fixed parameter here. By standard dynamic programming arguments, $V(x, y)$, $x \in \overline{O}$, y as above, is also the value function for the "classical" control problem of controlling Equation (10.5) on $[0, \nu]$ with cost

$$\int_0^\nu e^{-at} k_{i_0}(x(t), y, u(t)) \, dt + e^{-a\nu} h(x(\nu), y),$$

where $h(\cdot, \cdot) \equiv V(\cdot, \cdot)$ on ∂O. It follows that $V(x, y)$, $(x, y) \in O$ is the viscosity solution of the Hamilton-Jacobi equation for this problem [52], i.e., it must satisfy (in the sense of viscosity solutions) the p.d.e.

$$\min_u \left\{ \langle \nabla_x V(x, y), f_{i_0}(x, y, u) \rangle - a V(x, y) + k_{i_0}(x, y, u) \right\} = 0 \tag{10.6}$$

in O and hence on M^o. (Here ∇_x denotes the gradient in the x variable.) Elsewhere, standard dynamic programming heuristics suggest that Equation (10.6) holds with '=' replaced by '$\leq$'.

Based on the foregoing discussion, we propose the following system of **generalized quasi-variational inequalities** for $V(\cdot, \cdot)$: For $(x, y) \in X_i \times X_i$,

$$V(x, y) \leq \min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x,z)} V(z, z) \right\} \text{ on } C \tag{10.7}$$

$$V(x, y) \leq \min_v \left\{ c_a(x, v) + e^{-a\Delta_a(x,v)} V(G(x, v), G(x, v)) \right\} \text{ on } \sqcup_i (A_i \times X_i) \tag{10.8}$$

$$\min_u \left\{ \langle \nabla_x V(x, y), f_i(x, y, u) \rangle - a V(x, y) + k_i(x, y, u) \right\} \leq 0 \tag{10.9}$$

$$\begin{aligned} &\left(V(x, y) - \min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x,z)} V(z, z) \right\} \right) \\ &\quad \cdot \left(\min_u \left\{ \langle \nabla_x V(x, y), f_i(x, y, u) \rangle - a V(x, y) + k_i(x, y, u) \right\} \right) = 0 \text{ on } C \end{aligned} \tag{10.10}$$

(Equation (10.10) states that at least one of Equation (10.7), Equation (10.9) must be an equality on C.) (Equations (10.7)–(10.10) generalize the traditional **quasi-variational inequalities** encountered in impulse control [14]. We do not address the issue of well-posedness of Equations (10.7)–(10.10).

The following "verification theorem," however, can be proved by routine arguments.

Theorem 10.17 *Suppose Equations (10.7)–(10.10) has a "classical" solution V which is continuously differentiable "from the right" in the first argument and continuous in the second. Suppose $x(\cdot)$ is an admissible trajectory of our control system with initial data (x_0, y_0) and $u(\cdot)$, $\{v_i\}$, $\{\sigma_i\}$, $\{\zeta_i\}$, $\{\tau_i\}$, $\{\Gamma_i\}$ the associated controls and jump times, such that the following hold:*

1. *For a.e. $t \in \mathbb{T}$, i such that $x(t) \in X_i$,*

$$\langle \nabla_x V(x(t), x[t]), f_i(x(t), x[t], u(t)) \rangle + k_i(x(t), x[t], u(t)) =$$
$$\min_u \{ \langle \nabla_x V(x(t), x[t]), f_i(x(t), x[t], u) \rangle + k_i(x(t), x[t], u) \}.$$

2. *For all i,*

$$V(x(\sigma_i), x[\sigma_i]) = c_a(x(\sigma_i), v_i) + \exp\{-a\Delta_a(x(\sigma_i), v_i)\} V(G(x(\sigma_i), v_i), G(x(\sigma_i), v_i)).$$

3. *For all i,*

$$V(x(\zeta_i), x[\zeta_i]) = c_c(x(\zeta_i), x(\zeta_i')) + \exp\{-a\Delta_c(x, x(\zeta_i'))\} V(x(\zeta_i'), x(\zeta_i')).$$

Then $x(\cdot)$ is an optimal trajectory.

§10.4 DISCUSSION

The foregoing presents some initial steps towards developing a unified "state space" paradigm for hybrid control. Several open issues suggest themselves. We conclude with a brief list of some of the more striking ones.

1. A daunting problem is to characterize the value function as the unique viscosity solution of the generalized quasi-variational inequalities Equations (10.7)–(10.10).

2. Many of our assumptions can possibly be relaxed at the expense of additional technicalities or traded off for alternative sets of assumptions that have the same effect. For example, the condition $d(C_i, A_i) > 0$ could be dropped by having c_c penalize highly the controlled jumps that take place too close to A_i. (In this case, Assumption 10.4 has to be appropriately reformulated.)

3. Example 10.10 show that Assumption 10.3 cannot be dropped. In the autonomous case, however, the set of initial conditions that hit a C^∞ manifold are of measure zero [135]. Thus, one might hope that an optimal control would exist for almost all initial conditions in the absence of Assumption 10.3. The system of Example 10.11 showed this to be false. Likewise, in the systems of Example 10.10 we have, respectively, no optimal control for the sets

$$\{(x_1, x_2) \mid x_2 \le 0, x_1 < 1, x_2 + 1 > x_1\},$$

$$\{(x_1, x_2) \mid x_2 \le 0, x_1 < a, x_2 + a > x_1\} \cup \left([0,1] \times [-a/2, 0] - \overline{B([1, -1]^T, 1)}\right),$$

where $a = 2 - \sqrt{2}$ and $B(x, r)$ denotes the ball of radius r about the point x.

It remains open how to relax the conditions Assumptions 10.3 and 10.4. This might be accomplished through additional continuity assumptions on G, Δ_a, and c_a.

4. An important issue here is to develop good computational schemes to compute near-optimal controls, which is currently a topic of further research. See [51, §11.1] for some related work.

This is a daunting problem in general as the results of [25, §7] show that the hybrid systems models discussed in §3 can simulate arbitrary Turing machines (TMs), with state dimension as small as three. It is not hard to conceive of (low-dimensional) control problems where the cost is less than 1 if the corresponding TM does not halt, but is greater than 3 if it does. Allowing the possibility of a controlled jump at the initial condition that would result in a cost of 2, one sees that finding the optimal control is equivalent to solving the halting problem.

5. Another possible extension is in the direction of replacing X_{i_0} by smooth manifolds with boundary embedded in a Euclidean space. See [35] for some related work.

6. In light of Definition 10.13, all the proofs seem to hold if Assumption 10.1 is relaxed to only consider distances "from the right," that is if $\inf_i d_+(A_i, D_i) > 0$, with

$$d_+(A_i, D_i) \equiv \inf_{t>0, u(\cdot), x \in D_i} E_i^t(x, u(\cdot)) \in A_i,$$

where $E_i^t(x, u(\cdot))$ denotes the solutions under f_i with initial condition x and control $u(\cdot)$ in $U^{[0,t)}$. Here, time can be used as a "distance" in light of the uniform bound on the f_i; we consider $t > 0$ by adding that caveat that if we jump directly onto A_i, we do not make another jump until we hit it again. Presumably one must also make some transversality or continuity assumptions for well-posedness. This would allow the results to extend to many more phenomena, including those examples in [35].

§10.5 NOTES

The results of this chapter grew out of [105]. They are joint work with Profs. Vivek S. Borkar and Sanjoy K. Mitter completed in December 1993. They appeared in [32].

Hybrid control is a rapidly expanding field and we make no explicit reference to general papers here (see §3 and the references). Viable control, considered by [57, 86], was not discussed here. However, optimal control of hybrid systems has been considered in [95] (for the discrete-time case) and the pioneering work of [114]. Kohn is the first we know of to use relaxed controls and their ϵ-optimal approximations in the hybrid systems setting (see [114, Appendix I] and the references therein). The algorithmic importance of these was further described in [61].

We have already mentioned the link to impulse control. Even closer are the results of [154], discovered after this work was completed. That paper considers switching and "impulse obstacle" operators akin to those in Equations (10.7) and (10.8) for autonomous and (controlled) impulsive jumps, respectively. Yong *restricts the switching and impulse operators to be uniform in the whole space*, which is unrealistic in hybrid systems. However, he derives viscosity solutions of his corresponding Hamilton-Jacobi-Bellman system. His work may be useful in deriving viscosity solutions to our GQVIs.

Also after this work was completed, we became aware of the model and work of [152, §3.3]. In that paper, Witsenhausen considers an optimal, terminal constraint problem on

his hybrid systems model. Recall his model contains *no autonomous impulses, no controlled switching, and no controlled jumps.*

§10.6 APPENDIX

Proof. [of Theorem 10.8] Fix $x(0) = x_0 \in X_{i_0}$, $i_0 \in \mathbb{Z}_+$. Consider a sequence

$$(x^n(\cdot), u^n(\cdot), \{v_i^n\}, \{\sigma_i^n\}, \{\zeta_i^n\}, \{\tau_i^n\}, \{\Gamma_i^n\}), \qquad n \in \mathbb{Z}_+,$$

associated with our control system, with the obvious interpretation, such that $x^n(0) = x_0$ for all n and the corresponding costs decrease to $J(x_0)$. Let $y^n(\cdot)$ denote the solution of

$$\dot{y}^n(t) = f_{i_0}(y^n(t), x_0, u^n(t)), \qquad y^n(0) = x_0, n \in \mathbb{Z}_+. \tag{10.11}$$

Then $x^n(\cdot)$, $y^n(\cdot)$ agree on $[0, \tau_1^n)$. Since $\{f_i\}$ are bounded, $\{y^n(\cdot)\}$ are equicontinuous bounded in $C(\mathbb{R}_+; \mathbb{R}^{d_{i_0}})$, hence relatively sequentially compact by the Arzela-Ascoli theorem. The finite nonnegative measures $\eta^n(dt, dy) = dt\, u^n(t, dy)$ on $[0, T] \times U'$ are relatively sequentially compact in the topology of weak convergence by Prohorov's theorem [18]. $\{\tau_1^n\}$ are trivially relatively compact in $[0, \infty]$. Thus dropping to a subsequence if necessary, we may suppose that $y^n(\cdot) \to y^\infty(\cdot)$, $\eta^n(dt, dy) \to \eta^\infty(dt, dy)$, $\tau_1^n \to \tau_1^\infty$ in the respective spaces. Clearly η^∞ disintegrates as $\eta^\infty(dt, dy) = dt\, u^\infty(t, dy)$. Rewrite Equation (10.11) as

$$
y^n(t) = x_0 + \\
\left(\int_0^t f_{i_0}(y^n(s), x_0, u^n(s)) - f_{i_0}(y^\infty(s), x_0, u^n(s))\, ds \right) + \int_0^t f_{i_0}(y^\infty(s), x_0, u^n(s))\, ds
$$

for $t \geq 0$. By the uniform Lipschitz continuity of f_{i_0}, the term in parentheses tends to zero as $n \to \infty$. Since $\eta^n \to \eta^\infty$, the last term, in view of the relaxed control framework, converges to

$$\int_0^t f_{i_0}(y^\infty(s), x_0, u^\infty(s))\, ds$$

for $t \in [0, T]$. Since T was arbitrary a standard argument allows us to extend this claim to $t \in [0, \infty)$. (We use [18, Theorem 2.1(v), p. 12] and the fact that $\eta^\infty(\{t\} \times U') = 0$.) Hence $y^\infty(\cdot)$, $u^\infty(\cdot)$ satisfy Equation (10.11) with $n = \infty$. Since $d(C_{i_0}, A_{i_0}) > 0$, either $\tau_i^n = \sigma_i^n$ for sufficiently large n, or $\tau_i^n = \zeta_i^n$ for sufficiently large n. Suppose the first possibility holds. Then $y^\infty(\tau_1^\infty) = \lim x^n(\tau_1^n) \in A_{i_0}$. Let $v_i^n \to v_i^\infty$ in V along a subsequence. Then $c_a(x^n(\tau_1^n), v_1^n) \to c_a(y^\infty(\tau_1^\infty), v_1^\infty)$, $\Delta_a(x^n(\tau_1^n), v_1^n) \to \Delta_a(y^\infty(\tau_1^\infty), v_1^\infty)$, $\Gamma_1^n \to \Gamma_1^\infty \equiv \tau_1^\infty + \Delta_a(y^\infty(\tau_1^\infty), v_1^\infty)$. Set $x^\infty(\cdot) = y^\infty(\cdot)$ on $[0, \tau_1^\infty]$ and $x^\infty(\Gamma_1^\infty) = G(x^\infty(\tau_1^\infty), v_1^\infty)$. Then

$$\int_0^{\tau_1^n} e^{-at} k_{i_0}(x^n(t), x_0, u^n(t))\, dt \to \int_0^{\tau_1^\infty} e^{-at} k_{i_0}(x^\infty(t), x_0, u^\infty(t))\, dt. \tag{10.12}$$

If the second possibility holds instead, one similarly has $y_1(\tau_1^\infty) \in C_{i_0}$. Then Assumption 10.2 ensures that $\{x^n(\Gamma_1^n)\}$ is a bounded sequence in D and hence converges along a subsequence to some $y' \in D$. Then, on dropping to a further subsequence if necessary, $c_c(x^n(\tau_1^n), x^n(\Gamma_1^n)) \to c_c(y^\infty(\tau_1^\infty), y')$, $\Delta_c(x^n(\tau_1^n), x^n(\Gamma_1^n)) \to \Delta_c(y^\infty(\tau_1^\infty), y')$. Set $x^\infty(\cdot) = y^\infty(\cdot)$ on $[0, \tau_1^\infty]$, $\Gamma_1^\infty = \tau_1^\infty + \Delta_c(y^\infty(\tau_1^\infty), y')) = \lim \Gamma_1^n$ and $x^\infty(\Gamma_1^\infty) = y'$. Again Equation (10.12) holds. Note that in both cases, $x^\infty(\cdot)$ defined on $[0, \tau_1^\infty]$ is an admissible segment of a controlled trajectory for our system. The only way it would fail to be so is if

it hit A_{i_0} in $[0, \tau_1^\infty)$. If so, $x^n(\cdot)$ would have to hit A_{i_0} in $[0, \tau_1^n)$ for sufficiently large n by virtue of Assumption 10.3, a contradiction.

Now repeat this argument for $\{x^n(\Gamma_1^\infty + \cdot)\}$ in place of $\{x^n(\cdot)\}$. The only difference is a varying but convergent initial condition instead of a fixed one, which causes only minor alterations in the proof. Iterating, one obtains an admissible trajectory $x^\infty(\cdot)$ with cost $J(x_0)$. $\square$

Proof. [of Theorem 10.12] Recall the setup of Theorem 10.8. Consider the time interval $[0, \tau_1^\infty]$. Let $\overline{u}^n(\cdot)$, $n \in \mathbb{Z}_+$, be precise controls such that $dt\, \overline{u}^n(t, dy) \to \eta^\infty(dt, dy) = dt\, u^\infty(t, dy)$ in the topology of weak convergence. Let $\overline{y}^n(\cdot)$, $n \in \mathbb{Z}_+$, denote the corresponding solutions to Equation (10.11). Now τ_1^∞ equals either σ_1^∞ or ζ_1^∞. Suppose the former holds. As in the proof of Theorem 10.8, we have $\overline{y}^n \to y^\infty(\cdot)$ in $C([0, \infty), X_{i_0})$. Using Assumption 10.3 as in the proof of Theorem 10.8, one verifies that

$$\overline{\sigma}_1^n \equiv \inf\{t \geq 0 \mid \overline{y}^n(t) \in A_{i_0}\} \to \sigma_1^\infty.$$

Thus for any $\delta > 0$, we can take n large enough such that

$$
\begin{aligned}
|\sigma_1^\infty - \overline{\sigma}_1^n| &< \delta, \\
\sup\{\|\overline{y}^n(t) - y^\infty(t)\| \mid 0 \leq t < \sigma_1^\infty \vee \overline{\sigma}_1^n\} &< \delta, \\
|\overline{\sigma}_1^n + \Delta_a(\overline{y}^n(\overline{\sigma}_1^n), v_1^\infty) - \Gamma_1^\infty| &< \delta.
\end{aligned}
$$

Set $\overline{x}^n(\cdot) = \overline{y}^n(\cdot)$ on $[0, \overline{\sigma}_1^n]$ and $\overline{x}^n(\overline{\sigma}^n + \Delta_a(\overline{x}^n(\overline{\sigma}_1^n), v_1^\infty)) = G(\overline{x}^n(\overline{\sigma}_1^n), v_1^\infty)$ (corresponding to control action v_1^∞). The latter may be taken to lie in the open δ-neighborhood of $x^\infty(\Gamma_1^\infty)$ by further increasing n if necessary. In case $\tau_1^\infty = \zeta_1^\infty$, one uses Assumption 10.4 instead to conclude that $\overline{y}^n(t_n') \in C_{i_0}$ for some t_n' in the δ-neighborhood of τ_1^∞ for n sufficiently large. Set $\overline{\zeta}_1^n = t_n'$, $\overline{x}^n(\cdot) = \overline{y}^n(\cdot)$ on $[0, \overline{\zeta}^n)$. By further increasing n if necessary, we may also ensure that

$$
\begin{aligned}
\{\overline{x}^n(t) \mid t \in [0, \overline{\zeta}_1^n)\} \cap A_{i_0} &= \emptyset, \\
\sup\{\|\overline{y}^n(t) - x^\infty(t)\| \mid 0 \leq t \leq \overline{\zeta}_1^n \wedge \overline{\zeta}_1^\infty\} &< \delta, \\
|\overline{\zeta}_1^n + \Delta_c(\overline{x}^n(\overline{\zeta}_1^n), x^\infty(\Gamma_1^\infty)) - \Gamma_1^\infty| &< \delta.
\end{aligned}
$$

Set

$$\overline{x}^n(\overline{\zeta}_1^n + \Delta_c(\overline{x}^n(\overline{\zeta}_1^n), x^\infty(\Gamma_1^\infty))) = x^\infty(\Gamma_1^\infty).$$

It is clear how to repeat the above procedure on each interval between successive jump times to construct an admissible trajectory $\overline{x}^n(\cdot)$ with cost within ϵ of $J(x_0)$ for a given $\epsilon > 0$. $\square$

Proof. [of Theorem 10.14] Let $(x_n, y_n) \to (x_\infty, y_\infty)$ from the right in $\bigsqcup_i(X_i \times X_i)$ and let $\overline{x}^n(\cdot)$, $n \in \mathbb{Z}_+ \cup \{\infty\}$, denote optimal trajectories for initial data (x_n, y_n) respectively. By dropping to a subsequence of $n \in \mathbb{Z}_+$ if necessary, obtain as in Theorem 10.8 a limiting admissible trajectory $x'(\cdot)$ for initial data (x_∞, y_∞) with cost (say) α such that $V(x_n, y_n) \to \alpha \geq V(x_\infty, y_\infty)$. Suppose $\alpha > V(x_\infty, y_\infty) + 3\epsilon$ for some $\epsilon > 0$. Starting from $x^\infty(\cdot)$, argue as in Theorem 10.12 to construct a trajectory $\tilde{x}^n(\cdot)$ with initial data (x_n, y_n) for n sufficiently large, so that the corresponding cost does not exceed $V(x_\infty, y_\infty) + \epsilon$. At the same time, $V(x_n, y_n) \geq \alpha - \epsilon > V(x_\infty, y_\infty) + 2\epsilon$ for n sufficiently large, which contradicts the fact that $V(x_n, y_n)$ is the optimal cost for initial data (x_n, y_n). The claim follows. $\square$

Chapter 11

Hybrid Control Algorithms and Examples

In this chapter we outline four approaches to solving the generalized quasi-variational inequalities associated with optimal hybrid control problems. Then we solve some illustrative problems in our framework.

§11.1 ALGORITHMS FOR OPTIMAL HYBRID CONTROL

In §5, we proposed a very general framework for hybrid control problems that was shown to encompass all hybrid phenomena considered in this thesis and all hybrid models reviewed from the literature. A specific control problem was studied in this framework, leading to an existence result for optimal controls. The *value function* associated with this problem was seen to satisfy a set of *generalized quasi-variational inequalities* (*GQVIs*). In this section, we give explicit algorithms for computing the solutions to such optimal control problems.

There are two foundations to these algorithms. First, our unified view—treating continuous and discrete controls in a conceptually similar manner—led us to a generalized Bellman equation that may be solved via adaptations of standard methods: value and policy iteration.

Second, as previously noted [32, §10], the key to *efficient* algorithms for solving optimal control problems for hybrid systems lies in first establishing their strong connection to the models of impulse control and piecewise-deterministic processes (PDPs) [14, 51, 53, 156, §2.2.4, §2.2.5] Then, we modify the algorithms that have been found useful in solving *those* problems [50, 51]. The result is an impulse control-like algorithm and a linear programming solution for computing optimal controls.

The object of this section is to make the above observations more precise. Throughout we follow an amalgamation of the notation of [51, §10]. To ease notation we do not discuss the general case considered in §10, i.e., we assume V is only a function of the state (and not also the last jump point).

§11.1.1 BOUNDARY-VALUE ALGORITHMS

We now discuss explicit solutions of the GQVIs. First, note that instead of thinking of V defined on the generalized state-space in the previous section, $V : \bigsqcup_i X_i \to \mathbb{R}_+$, we may revert to considering it component-wise: $V_i : X_i \to \mathbb{R}_+$.

The Boundary-Value algorithms outlined in this section begin with a guess for each V_i. The algorithm is summarized as follows:

1. Given: a guess for each V_i.

2. For each i, compute the constrained boundary values arising from the GQVIs:

   (a) For $x \in A_i$ impose the constraint

   $$V_i(x) = \min_v \left\{ c_a(x, v) + e^{-a\Delta_a(x,v)} V(G(x, v)) \right\}$$

   (b) For $x \in C_i$ impose the constraint

   $$V_i(x) = \min \left\{ V_i(x), \ \min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x,z)} V(z) \right\} \right\},$$

3. Solve the GQVI Equation (10.9) to update each V_i separately, imposing the constraints of Step 2. This is an optimal control problem as we are used to encountering.

4. If convergence, exit; else go to Step 2.

Note that the boundary value problems in Step 3 can be solved in a variety of ways themselves, including value iteration, policy iteration, or (as in the case of linear systems) explicit methods [121]. As is usual in boundary-value problems, convergence in Step 4 is determined by comparing computed and constrained values. We have made no attempt to theoretically study the above algorithm for convergence, but we have used it to solve the hysteresis example discussed below.

§11.1.2 GENERALIZED BELLMAN EQUATIONS

Another solution method arises from our unified viewpoint. The algorithmic basis is the following Bellman Equation:

$$V^*(x) = \min_{p \in \Pi} \left\{ g(x, a) + V^*(x'(x, a)) \right\}, \tag{11.1}$$

where Π is a generalized set of actions. The three classes of actions available in our hybrid systems framework at each x are

- **Continuous Controls,** $u \in U$.

- **Autonomous Jumps,** possibly modulated by discrete controls $v \in V$ (if $x \in A$).

- **Controlled Jumps,** choosing source and destination (if $x \in C$).

To solve on a computer, we first (discretize the continuous state and controls and) compute for $x \in X_i$ the minimum of the following three quantities:

$$\min_u \left\{ \int_0^\delta k_i(x(t), u) \, dt + V \left(x + \int_0^\delta f_i(x(t), u) \, dt \right) \right\},$$

$$\min_{v \in V} \left\{ c_a(x, v) + e^{-a\Delta_a(x,v)} V(G(x, v)), \right\}$$

$$\min_{z \in D} \left\{ c_c(x, z) + e^{-a\Delta_c(x,z)} V(z) \right\},$$

with several caveats. First, we make sure that we do not hit A in our computation of the first quantity. (Briefly, if we do, we replace δ by $\sigma_{1,u} - t$.) Second, the last two quantities are not taken into account if $x \notin A$ or $x \notin C$, respectively.

The resulting system may be solved via value or policy iteration. So far, we have concentrated on value iteration (solved via relaxation) and call the resulting algorithm "Generalized Value Iteration."

More specifically, the algorithm is as follows:

1. Discretize the state space into $\mathcal{X}$ and set of continuous controls into $\mathcal{U}$.

2. For each $x \in \mathcal{X}$, guess a value for $V(x)$, e.g., zero.

3. For each $x \in \mathcal{X}$, compute the minimum of the three quantities described above, subject to the caveats. The first minimum is taken with respect to $u \in \mathcal{U}$ with the cost-to-go a weighted sum of the costs-to-go of nearest neighbors. Set $V(x)$ equal to that minimum. Repeat until convergence.

We do not go into theoretical results here, but only note that the theoretical framework of [45] is very general, allowing, e.g., piecewise Lipschitz continuous dynamics.

§11.1.3 ALGORITHMS INSPIRED FROM IMPULSE CONTROL

We now make more explicit the strong connection between hybrid systems and the models of impulse control and piecewise-deterministic processes (PDPs) §§2.2.4–2.2.5. More specifically, the controlled jumps sets of our model are precisely those sets in the impulse control framework where $\Gamma(x) \neq \emptyset$. The associated controlled jump costs are likewise completely analogous to those of impulse control. The connection with autonomous jump sets is only slightly less direct: they can viewed as "singular limits" of the jump measures λ and Λ of the the theory of PDPs. We have added the possibility of autonomous jump costs associated with such jumps. The jump destination sets are merely the union of the points that one can jump to using either controlled or autonomous jumps. We have also added autonomous and controlled jump transition delay maps, but these present no conceptual challenges, and are omitted from the following discussion.

One conceptual differences is the inclusion of continuous controls in our model. The other conceptual difference is that instead of having a single, continuous state space, e.g, $X = \mathbb{R}^n$, we have an indexed set of continuous state spaces, $X = \{X_i\}_{i=0}^{\infty}$ where each X_i is a subset of some Euclidean space $\mathbb{R}^{d_i}$, $d_i \in \mathbb{Z}_+$.

The splitting into discrete and continuous states is not a technical barrier. Indeed, the algorithms that arise from this structure are a natural consequence of the underlying "automata" interpretations (for example, our approximation techniques may be viewed as "unrolling" these automata into trees).

Now, we show how the algorithms in [51] may be modified to form the basis of control algorithms for hybrid systems. Generalizing from [51] to the case of our hybrid systems model, define the following operators:

$$
\begin{aligned}
\mathcal{J}(V_1, V_2)(t, x) = {} & \int_0^{\sigma_1 \wedge t} e^{-\alpha s} k(x(s)) \, ds \\
& + e^{-\alpha t} V_1(x(t)) 1_{\{t < \sigma_1\}} \\
& + e^{-\alpha \sigma_1} \min\{c_a(x(\sigma_1)) + V_2(G(x(\sigma_1)))\} \cdot 1_{\{\sigma_1 \leq t\}}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}V_2(x) &= \int_0^{\sigma_1} e^{-\alpha s} k(x(s)) \, ds + e^{-\alpha \sigma_1} \min\{c_a(x(\sigma_1)) + V_2(G(x(\sigma_1)))\} \\
&= \mathcal{J}(V_1, V_2)(\infty, x) \\
\mathcal{L}(V_1, V_2)(x) &= \left\{ \inf_{0 \le t < \sigma_1} \mathcal{J}(V_1, V_2)(t, x) \right\} \wedge \mathcal{K}V_2(x) \\
\mathcal{M}V_2(x) &= \inf_{y \in \Gamma(x)} \{c(x, y) + V_2(y)\} \\
\mathcal{A}V_2(x) &= \inf_{\tau \in M_\infty} \left\{ \int_0^t e^{-\alpha s} k(x_s) \, ds + e^{-\alpha t} \mathcal{M}V_2(x_t) \right\} \\
h(x) &= \int_0^\infty e^{-\alpha s} k(x_s) \, ds
\end{aligned}
$$

Above, the set M_∞ is the set of all times we are in the controlled jump set C. Finally, define $\hat{L}(V) = \mathcal{L}(\mathcal{M}V, V)$.

The following results may then be derived analogously to the results in [51].

Proposition 11.1 h *is the smallest solution of*

$$
\begin{cases}
V &= \mathcal{K}V \\
V &\ge 0
\end{cases}
$$

and $h_0 \equiv 0$, $h_{n+1} = \mathcal{K}h_n$ *implies* $h_n \uparrow h$ *as* $n \to \infty$.

Proposition 11.2 V^* *is the biggest solution of* $V = \mathcal{A}V$ *and* $V^* = \lim_{n \to \infty} \mathcal{A}^n h$.

Proposition 11.3 V^* *is the biggest solution of*

$$
\begin{cases}
V &= \hat{L}(V) \\
V &\le h
\end{cases}
$$

and $V_0 = h$, $V_{n+1} = \hat{L}(V_n)$ *implies* $V_n \downarrow V^*$ *as* $n \to \infty$.

Proposition 11.4 *Assume* $V_0 \ge h$, $V_{n+1} = \hat{L}(V_n)$. *Then* $\lim_{n \to \infty} V_n(x) = V^*(x)$, *for all* x.

We can thus use the $\hat{L}$ iterations above to compute the value function solving our GQVIs. Continuous controls u are taken care of by discretizing the control space into a finite number of values, treating the current value of u as part of an augmented state, and assigning a switching cost for transition between different control values. So, we perform $\hat{L}$ iterations on this problem. If performance of the resulting is satisfactory, we conclude. Else, we may decrease the cost of switching among continuous controls or increase the number of quantized control values considered. We refer to the above algorithm as the $\hat{L}$-Iteration algorithm.

Results on convergence of discretization of algorithms should follow similarly to those for impulse control [51].

§11.1.4 LINEAR PROGRAMMING SOLUTIONS

Linear programming has been used to solve for the value function associated with optimal control of Markov decision processes [124]. It has also been shown to be valuable in impulse control of PDPs [50]. Here we use it to solve hybrid control problems.

Again assume we have discretized the state space into $\mathcal{X}$ and the set of continuous controls into $\mathcal{U}$ (yielding $|\Pi|$ finite). Consider the right-hand side of Equation (11.1) as an operator, T, on functions V. Arguing as in [124, Lemma 6.21, p. 151], monotonicity of T and $V^* = TV^*$, imply that V^* may be obtained by solving

Maximize V subject to $TV \geq V$.

However, since maximizing $V(x)$ for each $x \in \mathcal{X}$ also maximizes $\sum V(x)$, the problem reduces to

Maximize $\sum V(x)$ subject to
$\min_{a \in \Pi} \{g(x,a) + V(x'(x,a))\} \geq V(x)$, for all $x \in \mathcal{X}$.

Alternatively, we can write this as

Maximize $\sum V(x)$ subject to
$\{g(x,a) + V(x'(x,a))\} \geq V(x)$, for all $a \in \Pi$ and $x \in \mathcal{X}$.

This is just a linear program and may be solved by standard techniques. The problem is that these linear programs may be large. If $|\mathcal{X}| = N$, and $|\Pi| = M$, then we have N variables and MN constraints. The advantage is that the matrix of constraints is sparse. Briefly, this is since the row corresponding to (x,a) only has non-zero weights for the nearest neighbors of the state $x'(x,a)$.

§11.2 CONTROLLING EXAMPLE SYSTEMS

We discuss below the solution of several simple, but illustrative control problems. We begin with a quick warmup. Then we discuss continuous control of a hysteresis system that exhibits autonomous switching. Then we discuss a satellite control problem. The on-off nature of the satellite's reaction jets creates a system involving controlled switching. We end with a transmission problem. The goal is to find the strategy of continuous accelerator input and discrete gear-shift position to achieve maximum acceleration.

WARMUP

First, going back to Example 10.11 we have

Example 11.5 *Consider Example 10.11 except with the controls restricted in $[-1, -\epsilon]$, $0 < \epsilon < 1$. Then the flows are transversal and do not vanish on $A_1 = \{0\}$ for any u. In this case, the optimal control exists. For example, if $K > 1/\epsilon$, one can show that $u(\cdot) \equiv -\epsilon$ is optimal.*

§11.2.1 HYSTERESIS EXAMPLE

We now consider a hybrid control example which combines continuous control with the phenomenon of autonomous switching.

Example 11.6 *Consider a control system with hysteresis:*

$$\dot{x} = f(x,u) \equiv H(x) + u,$$

where the multi-valued function H is shown in Figure 11-1.
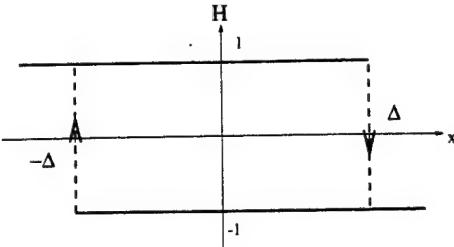
Figure 11-1: Hysteresis Function.

Note that this system is not just a differential equation whose right-hand side is piecewise continuous. There is "memory" in the system, which affects the value of the vector field. Indeed, such a system naturally has a finite automaton associated with the hysteresis function H, as pictured in Figure 11-2.
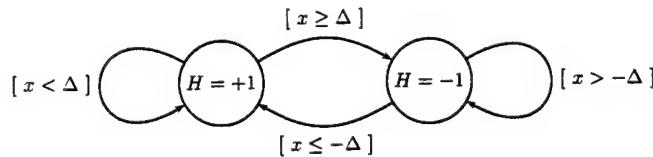
Figure 11-2: Finite Automaton Associated with Hysteresis Function.

As a control problem we consider minimizing

$$J = \int_0^\infty \frac{1}{2}(qx^2 + u^2)e^{-at}dt \equiv \int_0^\infty k(x,u)e^{-at}dt \qquad (11.2)$$

Let $s = H(x)$. We first solve for $V(x,s)$, and then for u. By symmetry, we expect $V(-\Delta, 1) = V(\Delta, -1)$. From the GQVIs, we expect V to satisfy

$$\begin{aligned} \min_u \{-aV(x,s) + V_x(x,s) \cdot f(x,u) + k(x,u)\} &= 0, \\ V(\Delta, 1) &= c + V(\Delta, -1), \\ V(-\Delta, -1) &= c + V(-\Delta, 1), \end{aligned} \qquad (11.3)$$

where s takes on the values ± 1 and c represents the cost associated with the autonomous-switchings.

We have solved these equations numerically (via the boundary-value algorithm) for the case $c = 0$, $a = 1$, $\Delta = 0.1$. The boundary-value algorithm outlined in §11.1 was used. The resulting control u, plotted against x (for $s = 1$) and q, is shown in Figure 11-3. As the state is increasingly penalized, the control action increases in such a way to "invert" the hysteresis function H.

To get an idea of the dynamics we plot the state and control over time for several values of q in Figure 11-4 and Figure 11-5, respectively. As the penalty on the state increases, the control acts in such a manner as to keep the state closer to zero for successively larger fractions of each "cycle." Concomittantly, we see how the controls begin to try to "invert" the hysteresis, as we just mentioned.

Optimal Policy vs. x and q
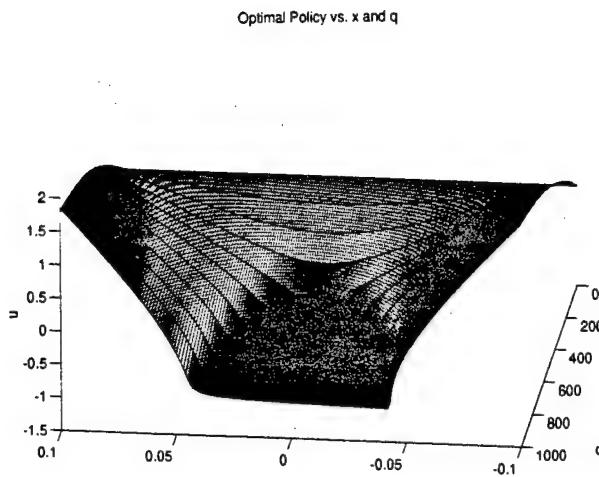
Figure 11-3: Optimal control u versus x and q, for the case $a = 1$, $c = 0$, $\Delta = 0.1$.
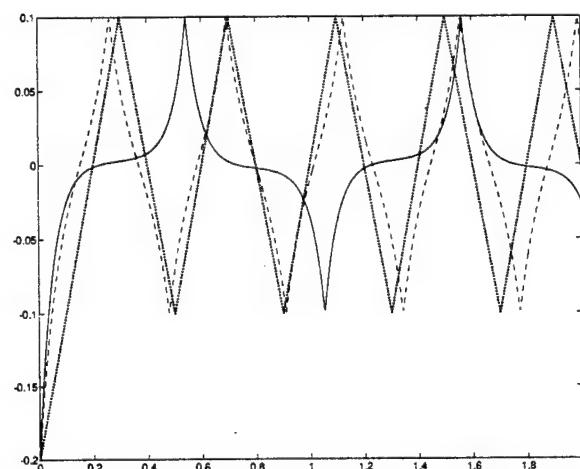
Figure 11-4: Comparison of x versus time, under different values of q. Solid, $q = 400$; dashed, $q = 200$; dotted, $q = 0$.
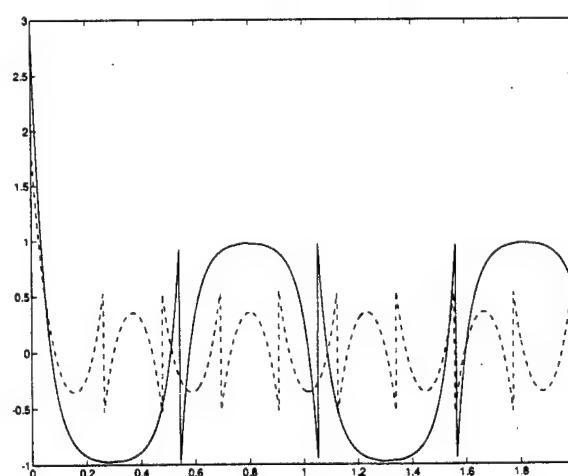
Figure 11-5: Comparison of u versus time, under different values of q. Solid, $q = 400$; dashed, $q = 200$; when $q = 0$, u is identically zero (not plotted)

Next, we compare the solutions found above with those computed by the generalized value iteration and linear programming algorithms of §11.1. In Figure 11-6 we plot the optimal control again, but this time we compare those previously shown (computed using the boundary-value algorithm) with those obtained using generalized value iteration arising from the Bellman equation described in §11.1. While the surfaces do look similar, we also plot several "slices" for better comparison. Up to the discretization used in the value iteration algorithm, the plots are seen to agree.

Next, we compare the optimal costs computed by linear programming with those computed with value iteration for successively tighter convergence criteria. See Figure 11-7. Note that the value iteration solutions are converging to those found via linear programming. Both algorithms used the same discretization and the same computer resources (Matlab on a Sparc5 after data initialization using a C program). In this case, the wall time to compute *each curve shown* was about equal. Thus for this problem linear programming yielded more than a factor of four speed-up.
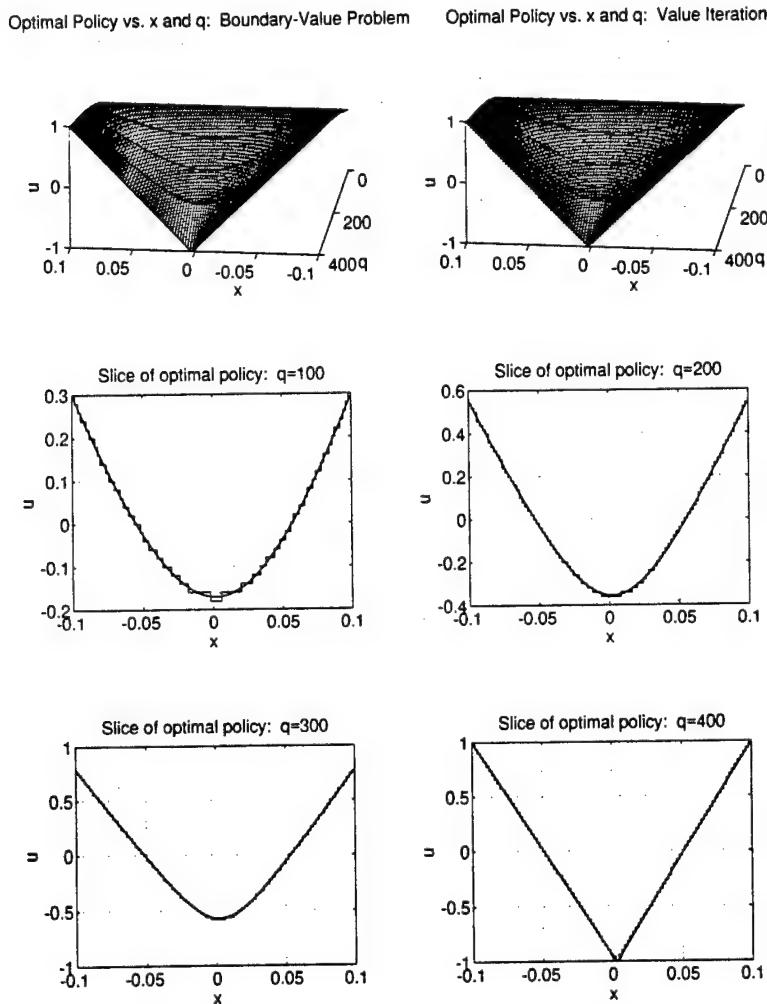
Figure 11-6: Comparison of policies computed by boundary-value and value iteration. Slices show details. Smooth, boundary-value; staircase, value iteration
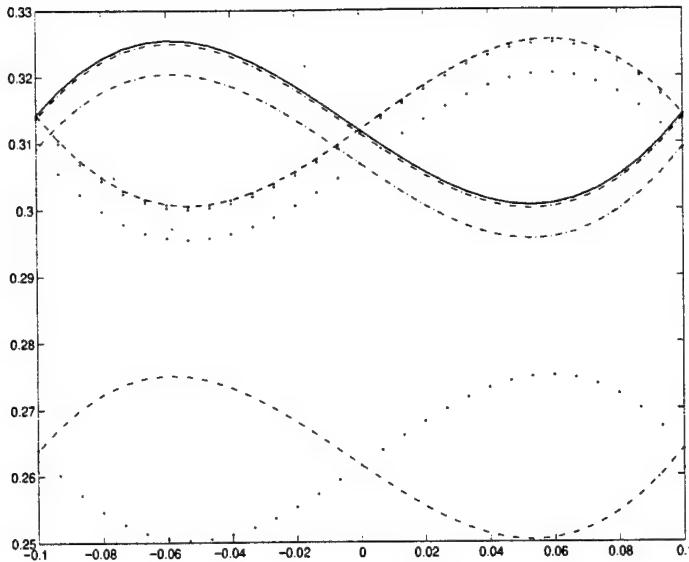
Figure 11-7: Cost-to-go for hysteresis example versus state ($s = 1, -1$ superimposed). Plot shows the convergence of value iteration solutions (dash-dot and dotted lines) to that found by linear programming (solid and dashed).

§11.2.2 EXAMPLES WITH CONTROLLED SWITCHING

We have also considered two problems involving controlled switching, a satellite station-keeping problem and a transmission control problem. The second also involves continuous controls.

SATELLITE CONTROL

Recall Example 3.3. It is a system with controlled switching. We add switching costs of c_{on} to switch from, and c_{off} to switch to, $v = 0$. We also penalize control and state with a running cost of $k = c\|v\| + x_1^2$.

We have solved the above using generalized policy iteration on Equation (11.1). We discretize the state-space and use the techniques described in [44] for dealing with edge effects. The resulting solution confirms the banded structure of positive, zero, and negative controls, that is typical for such problems. The reasoning is that there are long periods of drift followed by short bursts of control [145]. See Figure 11-8.

TRANSMISSION EXAMPLE

We now consider the transmission system of Example 3.4. We use $a(y) = \tanh(y/10)$ to represent our "torque curve." As a control problem we consider minimizing the running cost

$$k = (40 - x_2)^2,$$

starting from $x_1 = x_2 = 0$. One would expect that this leads to an optimal acceleration strategy for reaching a velocity of 40 as quickly as possible. We solved the problem using both value iteration and linear programming as outlined above. Figure 11-9 shows the
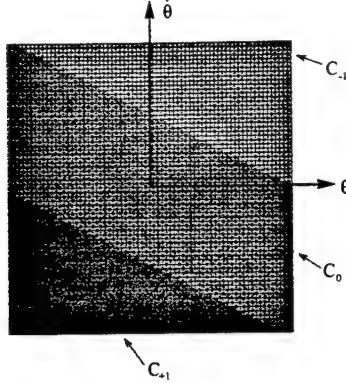
Figure 11-8: Satellite Station-Keeping Policy. State in C_q, denotes the policy is to use $v = q$.

results. The piecewise shifting strategies are interpreted as follows. The continuous control u is given by the fractional part, and the gear shift position by the integer part plus 1.

The "analytic strategy" shown results in maximum acceleration (up to the discretization of velocity and control used). In this case, it is simply given (as a function of x_2) by

$$\arg \max_{u,v} f(x_2, u, v) \equiv [-a(x_2/v) + u]/(1 + v).$$

The "calculated strategy" was the result of solving the linear program described in the previous section using the running cost above. Except for the edge effect, the calculated and analytic shifting strategies agree up to the discretization used. This also confirms that the simple chosen cost was a good choice for approximating an optimal acceleration strategy.

Further, we also plot four more curves, namely, the integral of the controlled vector field

$$f(x_2^*(t), u^*(t), v^*(t)),$$

versus both time and velocity for both the calculated and analytic solutions. In both cases, the "performance" of the calculated and analytic solutions are nearly identical.
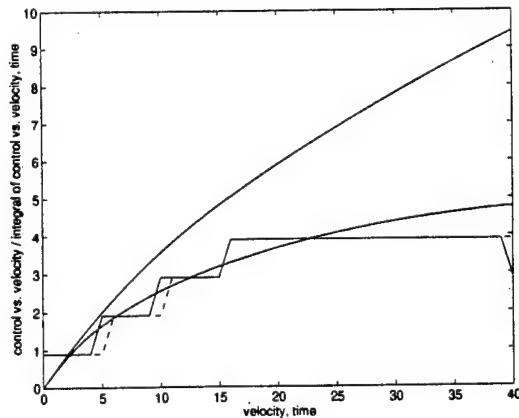
Figure 11-9: Optimal acceleration strategies and performance for the transmission problem. Solid, calculated; dashed, analytic.

Chapter 12

Conclusions and Future Work

This chapter summarizes the contributions of the thesis and gives a view toward future areas of research.

§12.1 CONCLUSIONS

This thesis studied different aspects of hybrid systems, concentrating in the broad areas of modeling, analysis, and control.

MODELING

Review. In §3 we first identified the types of discrete phenomena that arise in hybrid systems. Then we reviewed models of hybrid systems from the systems and control literature. We made some comparisons which enabled us to quickly prove simulation and modeling results for such systems.

Classification. §4 presented our own taxonomy of hybrid systems models. We introduced four main classes of systems:

- general hybrid dynamical systems

- hybrid dynamical systems, or hybrid systems,

- switched systems

- continuous switched systems

Both autonomous and controlled versions were introduced. Further classification was based on their structural properties and the discrete phenomena they exhibit. We also gave explicit instructions for computing the orbits and trajectories of general hybrid dynamical systems, including sufficient conditions for existence and uniqueness.

Unified. In §5 we formulated our own unified framework for hybrid systems modeling and control. We explicitly demonstrated that our unified model encompasses the identified discrete phenomena and the reviewed models of hybrid systems. The controlled version of the proposed model contains discrete and continuous states, dynamics, and controls. It is useful in posing and solution of hybrid control problems, including ones with disturbances. This was examined in Part III.

ANALYSIS

Topology. In §6 we discussed some of the topological issues that arise when differential equations and finite automata interact in hybrid systems. We concentrated on the maps from a continuum to a finite symbol space—AD maps—and back to another continuum—DA maps. We illuminated the general difficulties with the usual topologies in allowing continuous AD maps, constructed several topologies which bypassed them, and examined at length one such topology due to Nerode and Kohn.

We showed that there are inherent limitations present when one desires continuous maps from continuum to continuum through a finite symbol space, viz., one must equip the continua with new topologies. We constructed a control space topology allowing a continuous map which "completes the loop."

We ended with a different view of hybrid systems that may broach these problems. As an example, we showed that the most widely used fuzzy logic control structure is related to this different view and that it indeed is a continuous map from measurements to controls. We further demonstrated that these fuzzy logic controllers are dense in the set of such continuous functions.

We also made some connections with hybrid system trajectories and definitions of simulation.

Complexity. In §7 we explored the simulation and computational capabilities of hybrid systems. To accomplish this, we first defined notions of simulation of a discrete dynamical system by a continuous dynamical system. S-simulation, or simulation via section, was motivated by the definition of global section in dynamical systems [129]. Relaxing this to allow different parameterizations of time we considered P-simulation (simulation via points), which was seen to be weak. To remedy this, we defined I-simulation, or simulation via intervals. Both S-simulation and I-simulation imply P-simulation. S-simulation and I-simulation are independent notions.

We then showed that hybrid systems models with the ability to implement an exact clock can simulate fairly general discrete dynamical systems. Namely, we demonstrated that such systems can S-simulate arbitrary reversible discrete dynamical systems defined on closed subsets of $\mathbb{R}^n$. These simulations require ODEs in $\mathbb{R}^{2n}$ with the exact clock as input. Each of the reviewed hybrid systems models can implement exact clocks. They require only the most benign hybrid systems: *two discrete states and autonomous switching.*

Later, we found that one can simulate arbitrary discrete dynamical systems defined on subsets of $\mathbb{Z}^n$ without the capability of implementing an exact clock. Instead, one can use an approximation to an exact clock, implemented with a one-dimensional Lipschitz ODE. The result is that we can perform SI-simulations (resp. I-simulations) using continuous (resp. Lipschitz) ODEs in $\mathbb{R}^{2n+1}$.

Turning to computational abilities, we saw that there are systems of continuous ODEs possessing the ability to SI-simulate arbitrary pushdown automata and Turing machines. Finite automata may be SI-simulated with continuous, Lipschitz ODEs. By SI-simulating a universal Turing machine, we concluded that there are ODEs in $\mathbb{R}^3$ with continuous vector fields possessing the power of universal computation.

Finally, we showed that hybrid systems are strictly more powerful than Lipschitz ODEs in the types of systems they can implement. For this, we used a nontrivial example: the famous asynchronous arbiter problem. First, we settled the problem in an ODE framework by showing one cannot build an arbiter with devices modeled by Lipschitz ODEs continuous

in inputs and outputs. Then, we showed that each of the reviewed models of hybrid systems can implement arbiters, even when their continuous dynamics are modeled by Lipschitz ODEs continuous in inputs and outputs. Again, such examples require only autonomous switching and two discrete states.

Analysis Tools. §8 detailed work on switched and continuous switched systems. Each arise from hybrid systems by abstracting away the finite dynamics. In the first part of §8, we developed general tools for analyzing continuous switching systems. For instance, we proved an extension of Bendixson's Theorem to the case of Lipschitz continuous vector fields. This gives us a tool for analyzing the existence of limit cycles of continuous switching systems. We also proved a lemma dealing with the continuity of differential equations with respect to perturbations that preserve a linear part. Colloquially, this *Linear Robustness Lemma* demonstrates the robustness of ODEs with a linear part. The lemma is useful in easily deriving some of the common robustness results of nonlinear ODE theory (as given in, for instance, [11]). It also becomes useful in studying singular perturbations if the fast dynamics are such that they maintain the corresponding algebraic equation to within a small deviation. We added some simple propositions that allowed us to do this type of analysis in §9.

In the second part of §8, we introduced "multiple Lyapunov functions" as a tool for analyzing Lyapunov stability of switched systems. The idea here is to impose conditions on switching that guarantee stability when we have Lyapunov functions for each system f_i individually. Also, iterative function systems were presented as a tool for proving Lagrange stability and positive invariance. We also address the case where the finite index set is replaced by an arbitrary compact set.

Analyzing Examples. In §9.2 we presented an example hybrid control problem: the max system. This system was inspired from one used in the control of modern aircraft. The control law uses a logical function (max) to pick between one of two stable controllers: one a servo that tracks pilot inputs, the second a regulator about a fixed angle of attack. Typically, engineers resort to extensive simulation of even such simple systems because the analysis is too hard with their present toolbox. However, we analyzed the stability of this hybrid system in the case where the pilot input is zero and the controllers are linear full-state feedback. We showed that no limit cycles exist by proving and applying an extension of Bendixson's Theorem to the case of Lipschitz continuous vector fields; we also gave a Lyapunov function that proved all systems of this form are globally asymptotically stable. Interestingly, the Lyapunov equation used a logical switching.

In §9.3 we presented an analysis of a "continuation" of the max system. That is, we used a differential equation to obtain a "smooth" function instead of using the output given by the max function directly. By extending a result in the theory of continuity of solutions of ordinary differential equations, we proved stability properties of the continuation from those of the original max system.

The conclusion in this case is that the continuation method worked in reverse, i.e., it was easier to prove stability of the original, hybrid system directly. Furthermore, we concluded stability of the continuation via that of the original system. In effect, we showed robustness of the max system to the considered class of dynamic continuations.

CONTROL

Theoretical Results. In this part of the thesis, we took an optimal control approach to hybrid systems. In §10 we defined an optimal control problem in our unified hybrid control framework and derived some theoretical results. The problem, and all assumptions used in obtaining the remaining results, were expressly stated. The necessity of these assumptions—or ones like them—was demonstrated with examples. The main results were as follows: The existence of optimal and ϵ-optimal controls for the problem is established in §10.2. §10.3 gave a formal derivation of the associated generalized quasi-variational inequalities.

Algorithms and Examples. Using the GQVIs as a starting point, §11.1 concentrated on algorithms for solving hybrid control problems by solving the associated GQVIs. Our unified view led to the concept of examining a generalized Bellman equation. We also drew explicit relations with impulse control of piecewise-deterministic processes. Four algorithms were outlined: boundary-value algorithm, generalized value iteration, an impulse control-like approach, and linear programming.

Three illustrative examples were solved. We first considered a hysteresis system that exhibits autonomous switching and has a continuous control. Then we discussed a satellite station-keeping problem involving controlled switching. We ended with a transmission problem with continuous accelerator input and discrete gear-shift position. In each case, the optimal controls produced verify engineering intuition.

§12.2 FUTURE WORK

We have certainly seen a broad range of application areas to hybrid systems. We have only opened the door a little wider to further investigation. This should proceed along three fronts simultaneously: theory, applications, and development of engineering tools.

Modeling. One needs to explore the plethora of modeling choices available in hybrid systems. Since hybrid systems include dynamical systems as a subset, subclasses which permit efficient simulation, analysis, and verification should be explored. We believe that such a program is indeed being carried out by the computer scientists. Control theorists should do the same in their field in examining the hybrid control of hybrid systems.

Analysis. First, it is not hard to generalize our Multiple Lyapunov function (MLF) theory to the case of different equilibria, which is generally the case in hybrid systems (see §8.7). For example, under a Lyapunov-like switching rule, after all controllers have been switched in at level α_i, the set $\bigcup_i V_i^{-1}(\alpha_i)$ is invariant. Such a generalization is useful in hybrid systems, where different equilibrium generally arise. In this case, the multiple Lyapunov approach appears useful in establishing the convergence of optimization algorithms which perform jumps upon entering certain regions of the state-space [136]

Since sufficient conditions for stability can become guidelines for synthesis, control design arising from the constraints of §8 is a topic of further research. Following our example above, it can lead to the design of convergent algorithms.

There are also theoretical issues to be explored. Some examples include the stability of systems with multiple equilibrium points, the stability of switched systems, relations between fixed-point theory and Lyapunov stability, and the stability and dynamics of ordinary differential equations driven by Markov chains whose transition probabilities are a

function of the continuous state. The latter may provide a link to the large literature on jump systems (see §2.2.2).

Another important topic of further research is to incorporate developed analysis tools into software engineering tools. This will allow application of these tools to complicated examples in a timely manner.

Control. Specific open theoretical issues were discussed in §10.4. Another has to do with the robustness of our hybrid controls with respect to state. Here, our transversality assumptions should combine with Tavernini's result on continuity with respect initial condition to yield continuity of control laws on an open dense set.

An important area of current research is to develop good computational schemes to compute near-optimal controls in realistic cases. Analysis of rates of convergence of discretized algorithms should be explored. Later, the development of software tools to design such controllers automatically will become an important area of research.

On to Design. Finally, from modeling, through analysis and control, we come to design of complex, hybrid systems. Here, some of the interaction between levels is under our jurisdiction. What would we do with such freedom, coupled with our new-found analysis and control techniques? For example, we might design a flexible manufacturing system that not only allows quick changes between different product lines, but allows manufacturing of new products on the line with relative ease.

Consider the so-called reflex controllers of [116, 148], which constitute a dynamically consistent interface between low-level servo control and higher-level planning algorithms that ensures obstacle avoidance. Thus as a step in the direction of hierarchical, hybrid design, the reflex control concept is an example of how to incorporate a new control module to allow rapid, dynamically transparent design of higher-level programs. Further, there are some structures for the control programs used in research aircraft [79] that may lend themselves to such an approach. In each case, these designs incorporate structures which allow engineers to separate the continuous and logical worlds.

These controllers provide the inspiration, our analysis and control results the foundation, and our steps toward efficient algorithms the impetus, for setting a course toward design of hybrid systems. Ultimately, it is hoped they will lead to truly intelligent engineering systems.

Appendix A

Topology Review

The following is only a quick review (and we do assume some basic point set topology, e.g, [126, Ch. 2]). For more details consult [62, 75, 113].

A **topological space** consists of a set X and a **topology on X**. A topology on X is a set $\mathcal{T}$ of designated subsets of X, called **open sets**, such that

- $\emptyset$, X are in $\mathcal{T}$,

- $\mathcal{T}$ is closed under arbitrary unions,

- $\mathcal{T}$ is closed under finite intersections.

EXAMPLE. $\mathcal{T}_I = \{\emptyset, X\}$ and $\mathcal{T}_D = 2^X$ are each topologies on X, known as the **indiscrete** and **discrete** topologies, respectively.

If Y is a subset of topological space $(X, \mathcal{T})$, the collection

$$\mathcal{T}_Y = \{Y \cap U \mid U \in \mathcal{T}\}$$

is a topology on Y called the **subspace topology**. With this topology, Y is called a **subspace** of X, and $U \subset X$ is **open in Y** (or open **relative to Y**) if it belongs to $\mathcal{T}_Y$.

A set $C \subset X$ is called **closed** if $X \backslash C$ is open. A set can be open, closed, both, or neither. Suppose $\mathcal{T}$ and $\mathcal{T}'$ are two topologies on a given set X. If $\mathcal{T}' \subset \mathcal{T}$, we say that $\mathcal{T}'$ is **finer** than $\mathcal{T}$, and that $\mathcal{T}$ is **coarser** than $\mathcal{T}'$.

If X is a set, a **basis** for a topology on X is a collection $\mathcal{B}$ of subsets of X (called **basis elements**) such that

1. Each $x \in X$ is contained in at least one element of $\mathcal{B}$.

2. If x belongs to the intersection of two basis elements B_1 and B_2, then there exists a basis element $B_3 \subset B_1 \cap B_2$ containing x.

A **subbasis** $\mathcal{S}$ for a topology on X is a collection of subsets of X whose union equals X. The **topology generated by basis $\mathcal{B}$ (subbasis $\mathcal{S}$)** is the collection of all unions of (finite intersections of) elements of $\mathcal{B}$ ($\mathcal{S}$).

EXAMPLE. For any set X, the collection of all one-point sets of X is a basis for the discrete topology on X. The **open rays** of an ordered set X, the collection

$$\{x \mid x > a\}, \quad \{x \mid x < a\}, \qquad a \in X,$$

are a subbasis for the **order topology** on X.

The function $f : X \to Y$ is **continuous** if for all V open in Y, $U = f^{-1}(V)$ is open in X; it is **surjective** if its image is Y, **injective** if it is one-to-one, and **bijective** if both of these hold; it is a **homeomorphism** if it is a continuous bijection with continuous inverse.

A topological space X is **connected** if it cannot be written as $X = A \cup B$, where A and B are open, nonempty, and disjoint. Two useful results are the following.

- X is connected if and only if X and $\emptyset$ are the only subsets of X which are both open and closed.

- The continuous image of a connected space is connected.

The following classification of topological spaces is common:

- T_0. Given two distinct points in a topological space X, at least one of them is contained in an open set not containing the other.

- T_1. Given two distinct points in a topological space X, each of them is contained in an open set not containing the other.

- T_2 or **Hausdorff**. Given two distinct points in a topological space X, there are disjoint open sets, each containing just one of the two points.

The following hold for spaces in which *one point sets are closed.*

- **Regular**. Given a point and a closed set disjoint from it, there exist disjoint open sets containing each of them.

- **Normal**. Given two disjoint closed sets, there exist disjoint open sets containing each of them.
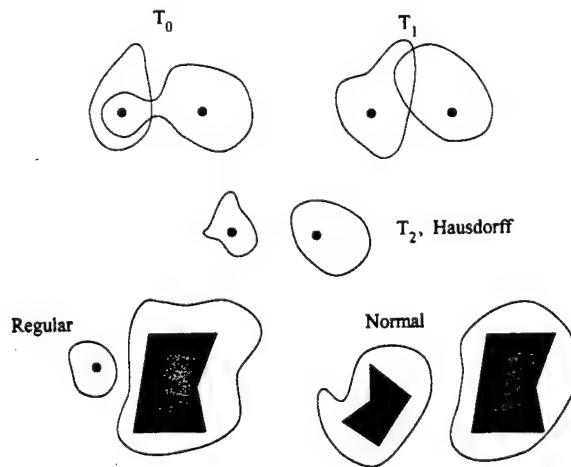
Figure A-1: Visualization of separation axioms.

A space Y is said to have the **universal extension property** if for each triple (X, A, f), where X is a normal space, $A \subset X$ is closed, and $f : A \to Y$ is continuous, there exists an extension of f to a continuous map of X into Y. For arbitrary index set J, $\mathbb{R}^J$ has the universal extension property [113, p. 216].

Bibliography

[1] R. Alur, C. Couroubetis, T. A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Grossman et al. [66], pages 209–229.

[2] R. Alur, A. Itai, R. P. Kurshan, and M. Yannakakis. Timing verification by successive approximation. *Information and Computation*, 118, 1995.

[3] Panos Anstaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors. *Hybrid Systems II*. Lecture Notes in Computer Science. Springer-Verlag, New York, 1995. To appear.

[4] Panos J. Antsaklis, Michael D. Lemmon, and James A. Stiver. Hybrid system modeling and event identification. Technical Report of the ISIS Group ISIS-93-002, ISIS Group, University of Notre Dame, 1993.

[5] Panos J. Antsaklis, James A. Stiver, and Michael D. Lemmon. Hybrid system modeling and autonomous control systems. In Grossman et al. [66], pages 366–392.

[6] Vladimir Igorevich Arnold. *Ordinary Differential Equations*. The MIT Press, Cambridge, MA, 1973.

[7] Eugene Asarin and Oded Maler. On some relations between dynamical systems and transition systems. Preprint, November 1993.

[8] Nils A. Baas. Hyper-structures as a tool in nanotechnology. *Nanobiology*, 3:49–60, 1994.

[9] Allen Back, John Guckenheimer, and Mark Myers. A dynamical simulation facility for hybrid systems. In Grossman et al. [66], pages 255–267.

[10] Dimitur D. Bainov and Pavel S. Simeonov. *Systems with Impulse Effect*. Ellis Horwood, Chichester, England, 1989.

[11] Evgenyii A. Barbashin. *Introduction to the Theory of Stability*. Wolters-Noordhoff, Groningen, The Netherlands, 1970.

[12] Michael Fielding Barnsley. *Fractals Everywhere*. Academic Press, Boston, 1988.

[13] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, November 1973.

[14] Alain Bensoussan and Jacques-Louis Lions. *Impulse Control and Quasi-Variational Inequalities*. Gauthier-Villars, Paris, 1984.

[15] Albert Benveniste and Gérard Berry, guest editors. *Proceedings of the IEEE*, 79(9), 1991. Special Issue on The Synchronous Approach to Reactive and Real-Time Systems.

[16] Albert Benveniste and Paul Le Guernic. Hybrid dynamical systems theory and the signal language. *IEEE Transactions on Automatic Control*, 35(5), 1990.

[17] N. P. Bhatia and G. P. Szegö. *Dynamical Systems: Stability Theory and Applications*, volume 35 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1967.

[18] Patrick Billingsley. *Convergence of Probability Measures*. Wiley, New York, 1968.

[19] Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1):1–46, July 1989.

[20] Leonard S. Bobrow and Michael A. Arbib. *Discrete Mathematics*. W. B. Saunders, Philadelphia, 1974.

[21] Michael S. Branicky. Analysis of continuous switching systems: Theory and examples. Technical Report LIDS-P-2215, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, November 1993.

[22] Michael S. Branicky. Equivalence of analog and digital computation. In *Workshop on Continuous Algorithms and Complexity*, Barcelona, October 1993. Centre de Recerca Matematica. Abstract.

[23] Michael S. Branicky. Stability of switched and hybrid systems. Technical Report LIDS-P-2214, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, November 1993.

[24] Michael S. Branicky. Topology of hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2309–2314, San Antonio, December 1993.

[25] Michael S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. Technical Report LIDS-P-2218, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, December 1993. Revised June 1994.

[26] Michael S. Branicky. Why you can't build an arbiter. Technical Report LIDS-P-2217, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, December 1993.

[27] Michael S. Branicky. Analog computation with continuous ODEs. In *Proc. IEEE Workshop Physics and Computation*, pages 265–274, Dallas, November 1994.

[28] Michael S. Branicky. Analysis of continuous switching systems: Theory and examples. In *Proceedings of the American Control Conference*, pages 3110–3114, Baltimore, June 1994.

[29] Michael S. Branicky. Continuity of ODE solutions. *Applied Mathematics Letters*, 7(5):57–60, 1994.

[30] Michael S. Branicky. Stability of switched and hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3498–3503, Lake Buena Vista, FL, December 1994.

[31] Michael S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, 1995.

[32] Michael S. Branicky, Vivek Borkar, and Sanjoy K. Mitter. A unified framework for hybrid control. Technical Report LIDS-P-2239, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, April 1994. Extended abstract in [33]. Summary and examples in [34].

[33] Michael S. Branicky, Vivek S. Borkar, and Sanjoy K. Mitter. A unified framework for hybrid control. In Cohen and Quadrat [47], pages 352–358. Extended Abstract.

[34] Michael S. Branicky, Vivek S. Borkar, and Sanjoy K. Mitter. A unified framework for hybrid control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4228–4234, Lake Buena Vista, FL, December 1994.

[35] Roger W. Brockett. Smooth multimode control systems. In Clyde Martin, editor, *Berkeley-Ames Conference on Nonlinear Problems in Control and Fluid Dynamics*, pages 103–110. Math-Sci Press, Brookline, MA, 1983.

[36] Roger W. Brockett. Smooth dynamical systems which realize arithmetical and logical operations. In H. Nijmeijer and J. M. Schumacher, editors, *Lecture Notes in Control and Information Sciences. Three Decades of Mathematical Systems Theory*, pages 19–30. Springer-Verlag, Berlin, 1989.

[37] Roger W. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and Its Applications*, 146:79–91, 1991.

[38] Roger W. Brockett. Hybrid models for motion control systems. In H. L. Trentelman and J. C. Willems, editors, *Essays in Control: Perspectives in the Theory and its Applications*, pages 29–53. Birkhäuser, Boston, 1993.

[39] Roger W. Brockett. Dynamical systems and their associated automata. In U. Helmke, R. Mennicken, and J. Saurer, editors, *Systems and Networks: Mathematical Theory and Applications*. Akademie Verlag, Berlin, 1994.

[40] Peter E. Caines and Yuan-Jun Wei. The hierarchical lattices of a finite machine. *Systems and Control Letters*, 1995. To appear.

[41] Christopher Chase, Joseph Serrano, and Peter J. Ramadge. Periodicity and chaos from switched flow systems: Contrasting examples of discretely controlled continuous systems. *IEEE Transactions on Automatic Control*, 38(1):70–83, 1993.

[42] Han-Fu Chen. Stochastic approximation and its new applications. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2–12, Proceedings of 1994 Hong Kong International Workshop on New Directions of Control and Manufacturing, 1994.

[43] Han-Fu Chen. Recent developments in stochastic approximation. Center for Intelligent Control Systems Seminar, Massachusetts Institute of Technology, May 1995.

[44] Herman Chernoff and John A. Petkau. A satellite control problem. Technical Report no. 9, Department of Mathematics, Massachusetts Institute of Technology, December 1977.

[45] Chee-Seng Chow and John N. Tsitsiklis. An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Transactions on Automatic Control*, 36(8):898–914, 1991.

[46] Moody T. Chu. On the continuous realization of iterative processes. *SIAM Review*, 30(3):375–387, September 1988.

[47] G. Cohen and J-P. Quadrat, editors. *Proceedings 11th INRIA International Conference on the Analysis and Optimization of Systems*, volume 199 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, New York, 1994.

[48] Thomas W. Colthurst. Advanced IFS. IAP course. Department of Mathematics, Massachusetts Institute of Technology, January 1993.

[49] M. Cosnard, M. Garzon, and P. Koiran. Computability properties of low-dimensional dynamical systems. In *Proc. 10th Symposium on Theoretical Aspects of Computer Science*, New York, 1993. Springer-Verlag. Lecture Notes in Computer Science, Volume 665.

[50] O. L. V. Costa. Impulse control of piecewise-deterministic processes via linear programming. *IEEE Transactions on Automatic Control*, 36(3):371–375, 1991.

[51] O. L. V. Costa and M. H. A. Davis. Impulse control of piecewise deterministic processes. *Math. Control Signals Syst.*, 2:187–206, 1989.

[52] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

[53] M. H. A. Davis. *Markov Models and Optimization*. Chapman and Hall, London, 1993.

[54] J. W. de Bakker, C. Huizing, W. P. de Roever, and G. Rozenberg, editors. *Real-Time: Theory in Practice*, volume 600 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1991.

[55] David F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Transactions on Automatic Control*, 35(8):916–924, August 1990.

[56] Akash Deshpande. *Control of Hybrid Systems*. PhD thesis, University of California at Berkeley, 1994.

[57] Akash Deshpande and Pravin Varaiya. Viable control of hybrid systems. In Anstaklis et al. [3]. To appear.

[58] Charles A. Desoer and Mathukumalli Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, New York, 1975.

[59] Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, 2nd edition, 1989.

[60] Jelel Ezzine and A. H. Haddad. Controllability and observability of hybrid systems. *International Journal of Control*, 49(6):2045–2055, June 1989.

[61] X. Ge, W. Kohn, and A. Nerode. Algorithms for chattering approximations to relaxed optimal controls. Technical Report 94-23, Mathematical Sciences Institute, Cornell University, April 1994.

[62] Michael C. Gemignani. *Elementary Topology*. Dover, New York, 2 edition, 1972.

[63] S. Di Gennaro, C. Horn, S. R. Kulkarni, and P. J. Ramadge. Reduction of timed hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4215–4220, Lake Buena Vista, FL, December 1994.

[64] Datta N. Godbole, John Lygeros, and Shankar Sastry. Hierarchical hybrid control: A case study. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1592–1597, Lake Buena Vista, FL, December 1994.

[65] Aleks Gollu and Pravin P. Varaiya. Hybrid dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2708–2712, Tampa, December 1989.

[66] Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors. *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1993.

[67] John Guckenheimer and Philip Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1990. 3rd Printing, Revised and Corrected.

[68] Nicolas Halbwachs. *Synchronous Programming of Reactive Systems*. Kluwer Academic, Boston 1993.

[69] David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.

[70] David Harel. Biting the silver bullet; toward a brighter future for system development. *IEEE Computer Magazine*, pages 8–20, January 1992.

[71] David Harel et al. STATEMATE: A working environment for the development of complex reactive systems. *IEEE Trans. Software Engineering*, 16(4):403–414, 1990.

[72] Zvi Har'El and Robert P. Kurshan. Software for analytical development of communications protocols. *AT&T Technical Journal*, January/February 1990.

[73] Morris W. Hirsch and Stephen Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, San Diego, 1974.

[74] Yu-Chi Ho, editor. *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*. IEEE Press, Piscataway, NJ, 1992.

[75] John G. Hocking and Gail S. Young. *Topology*. Dover, New York, 1988.

[76] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 1979.

[77] John J. Hopfield. The 1993 Herman Feshbach Lectures in Physics. Part I: The dynamics of computing in neural networks. Part II: Why biology looks strange to physicists. Part III: Modelling of olfactory computation. Department of Physics, Massachusetts Institute of Technology, April 1993.

[78] Chieh Su Hsu. *Cell-to-Cell Mapping*, volume 64 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1987.

[79] Charles S. Hynes. Flight control systems that can be formally verified. Technical report, NASA Ames Research Center, 1993. Extended abstract.

[80] Charles S. Hynes, George Meyer, Everett A. Palmer III, and Gordon H. Hardy. Vehicle management for the high speed civil transport. Technical report, NASA Ames Research Center, 1993. Extended abstract.

[81] i-Logix. The STATEMATE approach to complex systems. Technical report, i-Logix Inc., 1993. Technical product manual.

[82] Yuandong Ji and Howard Jay Chizeck. Controllability, stabilizability, and continuous-time Markovian jump linear quadratic control. *IEEE Transactions on Automatic Control*, 35(7):777–788, 1990.

[83] Timothy L. Johnson. Analytic models of multistage processes. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, December 1981.

[84] M. Joseph, editor. *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 331 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1988.

[85] Thomas Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

[86] W. Kohn, A. Nerode, J. Remmel, and A. Yaknis. Viability in hybrid systems. *Theoretical Computer Science*, 138(1), 1995.

[87] Petar Kokotovic, Hassan K. Khalil, and John O'Reilly. *Singular Perturbation Methods in Control: Analysis and Control*. Academic Press, London, 1986.

[88] A. N. Kolmogorov and S. V. Fomin. *Introductory Real Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1970. Rev. English ed. translated and edited by Richard A. Silverman.

[89] Renée Koplon and Eduardo Sontag. Sign-linear systems as cascades of automata and continuous variable systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2290–2291, San Antonio, December 1993.

[90] Sanjeev R. Kulkarni and Peter J. Ramadge. On the existence and complexity of convergent on-line decision rules. In *Proceedings of the IEEE Conference on Decision and Control*, New Orleans, December 1995. To appear.

[91] Petr Kůrka. Simulation in dynamical systems and Turing machines. Technical report, Department of Mathematical Logic and Philosophy of Mathematics, Charles University, Praha, Czechia, 1993.

[92] Huibert Kwakernaak and Raphael Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972.

[93] G. A. Lafferriere. Discontinuous stabilizing feedback using partially defined Lyapunov functions. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3487–3491, Lake Buena Vista, FL, December 1994.

[94] Y. Lin, E. D. Sontag, and Y. Wang. Recent results on Lyapunov-theoretic techniques for nonlinear stability. In *Proceedings of the American Control Conference*, Baltimore, June 1994.

[95] Jin Lu, Li-zhi Liao, Anil Nerode, and James H. Taylor. Optimal control of systems with continuous and discrete states. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2292–2297, San Antonio, December 1993.

[96] David G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications*. Wiley, New York, 1979.

[97] George Whitelaw Mackey. *The Mathematical Foundations of Quantum Mechanics*. Benjamin/Cummings, Reading, MA, 1980.

[98] Oded Maler, Zohar Manna, and Amir Pnueli. From timed to hybrid systems. In de Bakker et al. [54], pages 447–484.

[99] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag, New York, 1991.

[100] Leonard R. Marino. General theory of metastable operation. *IEEE Transactions on Computers*, 30(2):107–115, 1981.

[101] Michel Mariton. *Jump Linear Systems in Automatic Control*. Marcel Dekker, New York, 1990.

[102] Alexandre Megretsky. Robustness analysis of a relay system. Laboratory for Information and Decision Systems Seminar, Massachusetts Institute of Technology, April 1995.

[103] George Meyer. Design of flight vehicle management systems. In *Proceedings of the IEEE Conference on Decision and Control*, Lake Buena Vista, FL, December 1994. Plenary Lecture.

[104] Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Englewood Cliffs, NJ, 1967.

[105] Sanjoy K. Mitter and Vivek S. Borkar. 6.291; Special Topics: Optimal Control. Course Notes, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Fall 1993.

[106] Sanjoy K. Mitter and Shankar Sastry. 6.291; Special Topics: Intelligent Control. Course Notes, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Fall 1992.

[107] C. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64:2354–2357, 1990.

[108] C. Moore. Generalized shifts: Unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4:199–230, 1991.

[109] Frank Morgan. *Geometric Measure Theory*. Academic Press, Boston, 1988.

[110] A. S. Morse. Supervisory control of families of linear set-point controllers—Part 1: Exact matching. Preprint, March 1993.

[111] A. S. Morse. Supervisory control of families of linear set-point controllers—Part 2: Robustness. Preprint, November 1994.

[112] A. S. Morse. Control using logic-based switching. In A. Isidori, editor, *Trends in Control*, pages 69–113. Springer-Verlag, Berlin, 1995.

[113] James R. Munkres. *Topology*. Prentice-Hall, 1975.

[114] Anil Nerode and Wolf Kohn. Models for hybrid systems: Automata, topologies, stability. In Grossman et al. [66], pages 317–356.

[115] John Von Neumann and Oskar Morgenstein. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1953.

[116] Wyatt S. Newman and Michael S. Branicky. Experiments in reflex control for industrial manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1990.

[117] Louis Padulo and Michael A. Arbib. *System Theory; A Unified State-Space Approach to Continuous and Discrete Systems*. W. B. Saunders, Philadelphia, 1974.

[118] Theodosios Pavlidis. Stability of systems described by differential equations containing impulses. *IEEE Transactions on Automatic Control*, 12(1):43–45, 1967.

[119] Philippos Peleties and Raymond DeCarlo. Asymptotic stability of m-switched systems using Lyapunov-like functions. In *Proceedings of the American Control Conference*, pages 1679–1684, Boston, June 1991.

[120] A. Pnueli and J. Sifakis, guest editors. *Theoretical Computer Science*, 138(1), 1995. Special Issue on Hybrid Systems.

[121] William H. Press et al. *Numerical Recipes in C*. Cambridge University Press, Cambridge, England, 1988.

[122] Marc H. Raibert. *Legged Robots that Balance*. MIT Press, Cambridge, MA, 1986.

[123] Peter J. Ramadge. On the periodicity of symbolic observations of piecewise smooth discrete-time systems. *IEEE Transactions on Automatic Control*, 35(7):807–813, 1990.

[124] Sheldon M. Ross. *Applied Probability Models with Optimization Applications*. Dover, New York, 1992.

[125] Lee A. Rubel. A universal differential equation. *Bulletin of the American Mathematical Society*, 4(3):345–349, 1981.

[126] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3rd edition, 1976.

[127] Joseph La Salle and Solomon Lefschetz. *Stability by Liapunov's Direct Method With Applications*. Academic Press, New York, 1961.

[128] Louis Joseph Serrano. *The Effects of Time Sampling and Quantization in the Discrete Control of Continuous Systems.* PhD thesis, Princeton University, Department of Electrical Engineering, October 1990.

[129] Konstantin S. Sibirsky. *Introduction to Topological Dynamics.* Noordhoff International Publishing, Leyden, The Netherlands, 1975. Translated by Leo F. Boron.

[130] William McC. Siebert. *Circuits, Signals, and Systems.* MIT Press, Cambridge, MA, 1986.

[131] Hava T. Siegelman and Eduardo D. Sontag. Turing computation with neural nets. *Applied Math. Letters*, 4:77–80, 1991.

[132] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control.* Prentice-Hall, Englewood Cliffs, NJ, 1991.

[133] Eduardo D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, volume 6 of *Texts in Applied Mathematics.* Springer-Verlag, New York, 1990.

[134] Gunter Stein. Personal communication. March 1992.

[135] Lucio Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods, and Applications*, 11(6):665–683, 1987.

[136] Theodore Theodosopoulos. *Stochastic Models for Global Optimization.* PhD thesis, Massachusetts Institute of Technology, June 1995.

[137] Peter Murray Thompson. *Conic Sector Analysis of Hybrid Control Systems.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, August 1982.

[138] Tommaso Toffoli. Reversible computing. Technical Memo MIT/LCS/TM-151, Laboratory for Computer Science, Massachusetts Institute of Technology, February 1980.

[139] Vadim I. Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2):212–222, 1977.

[140] Pravin P. Varaiya. Smart cars on smart roads: Problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, February 1993.

[141] Steven Vickers. *Topology Via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, Cambridge, England, 1989.

[142] Mathukumalli Vidyasagar. *Nonlinear Systems Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1978. Also: 2nd Edition, 1993.

[143] Edward R. Vrscay. Iterated function systems: Theory, applications, and the inverse problem. In J. Bélair and S. Dubuc, editors, *Fractal Geometry and Analysis*, pages 405–468. Kluwer Academic Publishers, The Netherlands, 1991.

[144] Stephen A. Ward and Robert H. Halstead. *Computation Structures.* MIT Press, Cambridge, MA, 1989.

[145] Ronald Allen Warren. On-off satellite attitude control: An investigation. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering, 1967.

[146] Yuan-Jun Wei and Peter E. Caines. Hierarchical COCOLOG for finite machines. In Cohen and Quadrat [47], pages 29–38.

[147] Thomas S. Wickman. *Reflex Control for Robot System Preservation, Reliability, and Autonomy.* PhD thesis, Case Western Reserve University, Department of Electrical Engineering and Applied Physics, August 1994.

[148] Thomas S. Wickman, Michael S. Branicky, and Wyatt S. Newman. Reflex control for robot system preservation, reliability, and autonomy. *Computers and Electrical Engineering*, 20(5):391–407, 1994. Special Issue on Fault Tolerance and Robotics.

[149] Mark A. Wicks, Philippos Peleties, and Raymond A. DeCarlo. Construction of piecewise Lyapunov functions for stabilizing switched systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3492–3497, Lake Buena Vista, FL, December 1994.

[150] Jacques Leopold Willems. *Stability Theory of Dynamical Systems.* Nelson, London, 1970.

[151] David George Wimpey. *Finite-State Control of Discrete-Time Continuous Processes: An Automata Motivated Approach.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June 1982.

[152] Hans S. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2):161–167, 1966.

[153] John L. Wyatt. 6.243; Nonlinear Systems Analysis. Course Notes, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Spring 1991.

[154] Jiongmin Yong. Systems governed by ordinary differential equations with continuous, switching and impulse controls. *Applied Mathematics and Optimization*, 20:223–235, 1989.

[155] Laurence Chisolm Young. *Lectures on the Calculus of Variations and Optimal Control Theory.* Chelsea, New York, 2nd edition, 1980.

[156] Jerzy Zabczyk. Optimal control by means of switching. *Studia Mathematica*, 65:161–171, 1973.

[157] N. P. Zhidkov. Nekotorie svoistva diskretnikh dinamicheskikh sistem. *Moskovskii Gosudarstvennyi Universitet Lomonosova Uchenye Zapiski*, 163(Matematika 6):31–59, 1952. Russian.

Symbol Index

Common Notation

$\mathbb{R}$, $\mathbb{R}_+$	reals, nonnegative reals
$\mathbb{Z}$, $\mathbb{Z}_+$	integers, nonnegative integers
$X \backslash U$, $X - U$	complement of U in X
$\overline{U}$	closure of U
U^o	interior of U
∂U	boundary of U
$|A|$	cardinality of set A
$A \simeq B$	A, B are sets with $|A| = |B|$
$f(t^+)$, $f(t^-)$	right-, left-hand limits of f at t
$C(X, Y)$	continuous functions from X to Y
v^T	transpose of v
$\|x\|$, $\|x\|_2$, $\|x\|_\infty$	arbitrary, Euclidean, and infinity norm of x
∇f	divergence of f

Special Notation

$\underline{N}$	$\{1, 2, \ldots, N\}$
$\lfloor x \rfloor$	greatest integer less than or equal to x
$\lceil x \rceil$	least integer greater than x
$[t]$	time less than or equal to t at which the last jump occurred
$[t]_p$	time at which the variable p last jumped
q^+	successor of $q(t)$
q^-	predecessor of $q(t)$
[condition]	transition enabled (event is triggered) if condition is true
![condition]	transition must be taken (event must be accepted) if condition is true
?[condition]	transition may be taken (event may be accepted) if condition is true
$[X, S, f]$	dynamical/transition system f defined on X over the semigroup S
$\tilde{G}$	continuous extension of G
$\mathcal{T}$	topology
$\mathcal{B}$	basis
$\mathcal{S}$	subbasis

Hybrid Systems Models

State Spaces
S hybrid state space
X continuous state space
Q discrete state space

Designated Subsets
A autonomous jump set
C controlled jump set
D jump destination set
$M_{q,p}$ transition manifolds $(q, p \in Q)$

States
s hybrid state
x continuous state
q discrete state

Dynamics
ϕ_q extended transition functions
f_q transition functions
G jump transition maps
J impulse transition map (continuous component of G)
ν switching transition map (discrete component of G)

Transition Delays
Δ_a autonomous jump delay map
τ_i pre-jump times
Γ_i post-jump times
σ_i pre-jump times for autonomous jumps

Controls
u continuous control
v discrete control (exercised upon autonomous jumps)
F controlled jump transition map (set-valued)
Δ_c controlled jump delay map
ζ_i intervention (pre-jump) times for controlled jumps
ζ_i' post-jump times for controlled jumps

Costs
k running cost
c_a autonomous jump cost
c_c controlled jump cost
$V(\cdot)$ value function

Index